

Request for public comments

OSS Sample Curricula for Software Engineering Education - Skillsets and Sample Courses -

Draft 2.0

December 1, 2009

Northeast Asia OSS Promotion Forum WG2

China OSS Promotion Union
Japan OSS Promotion Forum
Korea OSS Promotion Forum

Each portion of this document is copyrighted
by member companies and/or individuals in
China OSS Promotion Union
Japan OSS Promotion Forum
Korea OSS Promotion Forum

This document is licensed under the Creative Commons License
with Attribution 2.0 international and later internationalized license
(CC-BY 2.1JP / CC-BY 2.0KR / CC-BY 2.5CN)

<http://creativecommons.org/international/>
<http://creativecommons.org/licenses/by/2.1/jp/>
<http://creativecommons.org/licenses/by/2.0/kr/>
<http://creativecommons.org/licenses/by/2.5/cn/>



All company, product, service names used in this document are for identification purposes only. All trademarks and registered trademarks are the property of their respective owners.

Table of Contents

1. Introduction	7
1.1 Terminology and definitions	7
1.2 Coding rules	9
1.3 Overall Sample Curricula Architecture	9
1.4 Industrial Masking Sheet & Personal Record Sheet	11
2. Detailed Description of Skill-sets.....	12
2.1 Knowledge of OSS.....	13
2.2 Kernel of Linux	22
2.3 Linux System Programming.....	28
2.4 Network Architecture	34
2.5 C	41
2.7 Script Languages	45
2.8 Web Programming.....	50
2.9 Fundamentals of Cryptography	61
2.10 Network Security.....	69
2.11 OS Security.....	75
2.12 Basic Skills in RDB.....	80
2.14 Embedded Systems Development.....	87
2.15 Embedded Development Environment	92
2.16 Device Driver Development.....	96
Appendix A. Sample Courses	102
A-1. Sample Course for OSS based Embedded SW Developer.....	102
A-2 Sample course for Web based Enterprise Application Developer by OSS	104
Appendix B (informative).....	106

Foreword

The NEAOSS Forum (Northeast Asia Open Source Software promotion Forum) was formed by the Chinese, Korean and Japanese governments and regional organizations for OSS promotion; the China OSS Promotion Union, Korea OSS Promotion Forum and Japan OSS Promotion Forum. The Forum intends to promote Open Source Software within northeast Asia.

The NEAOSS Forum formed “WG2: Human Resource Development” in order to promote and develop Human Resources for Open Source Software application and development in July 2004.

In accordance with the CJK DG Memorandum for the 4th (Tianjin), 5th (Fukuoka) and 6th (Seoul) NEA OSS Promotion Forums, the WGs have been discussing topics such as the mutual testing and certification of OSS experts, and the development of curriculum and textbooks for OSS developers and users.

The NEAOSS Forum WG2 formed a subsidiary group, Task Force 1, in April 2006 at the NEA OSS Promotion Forum in Tianjin. It addresses the research of the current status of Human Resource Development for Open Source Software in the northeast Asian region on both the demand and provider side and development of the NEA HRD Sample Curricula for colleges, lecturers, enterprises, and institutes to develop human resources with OSS skills.

WG2 delivered the “NEA HRD Analysis Report” (draft 1.0) in December 2007 and published the report ver.1.0 in September 2008. WG2 develops the NEA HRD Sample Curricula for OSS which is based on this research result.

As far as open source software (hereinafter called “OSS”) is concerned, there has been steady improvement to date in operating systems, middleware, networks, and development tools to build IT systems, thus reinforcing the status of OSS as a consistent software technology infrastructure to sustain our information economy. However, while there are growing needs for OSS as an important infrastructure, there is a severe shortage of engineers competent enough to use OSS, which is one of the greatest factors impeding the proliferation of the software. The survey on NEA OSS human resource development titled the “NEA HRD Analysis Report” pointed out a wide gap among businesses and IT service providers between expectations for OSS engineers and the real situation.

For OSS to achieve wide recognition for its usefulness and to spread throughout society, it is essential to cultivate IT engineers capable of building, operating, and maintaining OSS-based IT systems with sufficient knowledge of OSS. For that purpose, we are urged to establish sample curricula to be used as a reference in OSS technical training and to promote it among universities, vocational schools, IT training providers, and other organizations. Under these circumstances, this project resulted in the drafting of the NEA HRD Sample Curricula Version 2.0 after releasing version 1.0 on August 31, 2009 through the Website: <http://www.neaoss-forum.org> .

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. NEAOSS shall not be held responsible for identifying any or all such patent rights.

Contributors

Korea OSS Promotion Forum

Kern Koh, Seoul National University
Doohyun Kim, Konkuk University
Minsuk Lee, Hansung University
Sam Song, BIZ Kernel, Inc.
Young Jong Kim, Pumpkin Networks, Inc.
Young Gyun Oh, Rockplace, Inc.
Soon Young Jung, Korea University
Teak-Jong Kim, TmaxSoft
Je-Ho Kim, CDC Group, Inc.
Ki-Cheol Han, Darwin Consulting and Solution
Keun Dong Kim, Rockplace, Inc.

Japan OSS Promotion Forum

Ryusuke Kajiyama, Sun Microsystems Inc.
Hisashi Hashimoto, Hitachi Ltd.
Makoto Oya, Shonan institute of Technology
Yutaka gNiibe, National Institute of Advanced Industrial Science and Technology
Chieko Takahashi, LPI Japan
Eiichi Sato, Zend Japan, Ltd.
Kuniaki Hori, NEC Learning Ltd.
Kaori Inaba, SRA OSS Inc.
Katsuya Kobayashi, NTT COMWARE Corp.
Kazuo Hiyane, Mitsubishi Research Institute, Inc.
Seiichi Furuta, NTT DATA Corp.
Kenichiro Hamano, NTT DATA Corp.
Hiroshi Miura, NTT DATA Corp.
Hirotaka Yoshioka, Rakuten, Inc.
Koichi Oouchi, Information-processing Promotion Agency

Kei Masumoto, NTT DATA Corp.

Daisuke Numaguchi, NTT DATA Corp.

China OSS Promotion Union

Zhong Chen, Peking University

Qi Jing, Peking University

Qingni Shen, Peking University

Weiping Wen, Peking University

Lian Yu, Peking University

Huiping Sun, Peking University

Guorui Wu, Peking University

Xu Chen, Intel Corporation

Fleming Feng, Intel Asia-Pacific Research and Development Ltd.

Each portion of the document is copyrighted by the above members.

1. Introduction

1.1 Terminology and definitions

(1) Skills

In general, "skill" is often referred to as knowledge and expertise on element technology of a specific product, know-how on the application of service, and specific programming language, etc. In the field of OSS, skill means expertise and knowledge for following diverse, but not limited, activities:

- Creating and maintaining new OSS projects
- Updating existing OSS as contributors or committers
- Applying OSS to software development/system development
- Utilizing OSS for system construction/ management

(2) Skill Category

Skill category is a set of skills under the same technical domain. For example, Java, C/C++ and Web programming skills are element skills under the programming skill category. Table 1 shows OSS skill sets composed of 38 skills under the nine skill categories dealt with in this document.

(3) Topic, subtopic, and unit

A skill is defined by topic. A topic has subtopics. The topic is like a chapter in a textbook. For example, "building a cross development environment" is a topic under the "embedded S/W development environment" skill. Meanwhile, "able to use minicom" is a subtopic under the "building a cross development environment" topic.

The subtopics are grouped into three levels, but a topic does not need to have subtopics in all three levels. A group of subtopics in a level is called a unit. The unit is described with objective, prerequisite and unit code as shown in Table 1; consequently, it contains the most detailed items supporting a skill.

In some cases, a certain unit is equivalent to or inclusive of the other unit in a different topic in a different skill. Such a unit can be shared with the other corresponding unit. For example, the level-1 unit in the "kernel debugging" topic, whose subject is "Setup remote debugging environment and debug," in the "device driver development" skill can be included in the level-1 unit in the "remote debugging" topic of the "Embedded S/W Development Environment" skill.

Figure 1. OSS skill set (9 Skill Categories and 38 Skills)

Skill Category	Skill	Skill Category	Skill
Basic	Knowledge of OSS	Multimedia System	Multimedia Programming
	Legal Affairs		Multimedia Service Platform
	Computer System and Architecture	Development system	Development Frameworks
	Distributed Architecture		Development Tools
System	Concept of Linux and Basic Operations	Development system	Integrated Development Environment
			Software Testing
	Kernel of Linux	Security	Fundamental of Cryptography
			Network Security

	Linux System Management			
	Linux System Programming		OS Security	
	Network Server Management		RDB	Basic Skills in RDB
	Cluster System Architecture			RDB system management
	Concurrent System Programming			RDB Applications Development
	Java EE Application Server		Embedded SW	Embedded System Development
Network Architecture	Embedded Development Environment			
Network Management	Embedded Application Development			
Network Programming	Embedded System Optimization			
Programming	Java		Device Driver Development	
	C			
	C++			
	Script Language			
	GUI			
	Web Programming			

SKILL CATEGORY NAME	< Name of Skill Category >		SKILL CATEGORY NO.	<N>
SKILL NAME	< Name of the Skill >		SKILL NO.	<M>
TOPICS	LEVEL	DESPRICTIONS & SUBTOPICS		Topic Code
Topic 1 (has a level II topic with prerequisite >	I	-		-
	II	Objective	< Descriptive objective of the level of topic 1 >	N-M-1-II
		Prerequisite	< prerequisite topic code(s) >	
		- Subtopic 1 ■ Sub-subtopic . ■ ... - Subtopic 2 ■ ...		
III	-		-	
Topic a < has a code shared level I topic, and a level II topic with no prerequisite >	I	Objective	< Copied objective from the shared topic P-Q-r-I >	CODE SHARE P-Q-r-I
	II	Objective	< Descriptive objective of the level of topic a >	N-M-a-II
		- Subtopic 1 ■ Sub-subtopic . ■ ... - Subtopic 2 ■ ...		
		-		
III	-		-	
Topic b < Code shared topic Level I with X-Y-w-I, Level II with X-Y-w-II >	I	Objective	< copied objective from the shared topic X-Y-w-I >	CODE SHARE X-Y-w-I
		Prerequisite	< copied prerequisite codes from shared topic X-Y-w-I >	
	II	Objective	< copied objective from the shared topic X-Y-w-II >	CODE SHARE X-Y-w -II
	III	-		-

Figure 1. Skill definition form

(4) Skill proficiency and level

Skill proficiency is defined as a necessary degree of maturity for performing corresponding activities. Skill proficiency is described in the criteria as “able to do so,” and it is evidence of reaching a certain level of capability in a specific activity. In other words, it provides a guideline whereby an individual who is assessed at a specific level in an activity is able to perform a certain degree of work for a corresponding particular skill. The OSS human resource is indexed by three levels as follows:

- Level-1: As a professional, he/she understands the minimum common knowledge for information technology. He/she can work by utilizing IT knowledge for his/her job.
- Level-2: Working to realize an IT solution, product and service. He/she plays a role with advice from a superior.
- Level-3: Working to realize an IT solution, product and service, he/she fulfills a role on his own.

1.2 Coding rules

For the convenience in utilizing skill definitions, the unit shall have a unique code following simple and consistent rules. In this document, the unit is coded with the form of C-S-T-U, each element of which stands for the following:

- C: skill Category
- S: Skill
- T: Topic
- U: Unit

In the previous example, the “able to use minicom” is one of the subtopics in the Level-1 unit coded as 9-2-4-I, because the “Embedded SW” skill category, the “embedded S/W development environment” skill, and the “building a cross development environment” topic are numbered as 9, 2, and 4, correspondingly. Meanwhile, in the case of sharing a unit with another topic, the code of the corresponding unit is used. For example, the Level-1 unit of the “kernel debugging” topic can have the same code as this 9-2-4-I, because the objective is the same.

1.3 Overall Sample Curricula Architecture

As shown in Figure 2, the course is a set of classes which have precedence relationships, and a class is a collection of units in a topic. For example, while class 2 of course 2 is organized with only the Level-2 unit (1-2-1-II) of topic number 1-2-1, class 1 of course M is composed of two units, i.e., 1-2-1-II and 1-2-1-III. Whether a unit is contained in the class of a certain course or not depends on the course designer’s purpose. Similarly, by including objectives, outcomes, prerequisites, modules, and hours in the syllabus, the organization of a class is wholly dependent on the course designer.

On the other hand, classes can be represented by corresponding topics because the skill sets can be applied to a course with scalability. While Figure 2 shows a case of a course with subtopic-level granularity, Figure 3 shows a case with topic-level granularity.

The course designer can be typically a group of persons, as well as one person, from industrial sectors, public educational and training sectors, and/or in-house training sectors. They may first survey the status of the OSS human resource demands in a specific job category, and seek a solution by designing curriculum, a set of courses, to properly meet the survey demands.

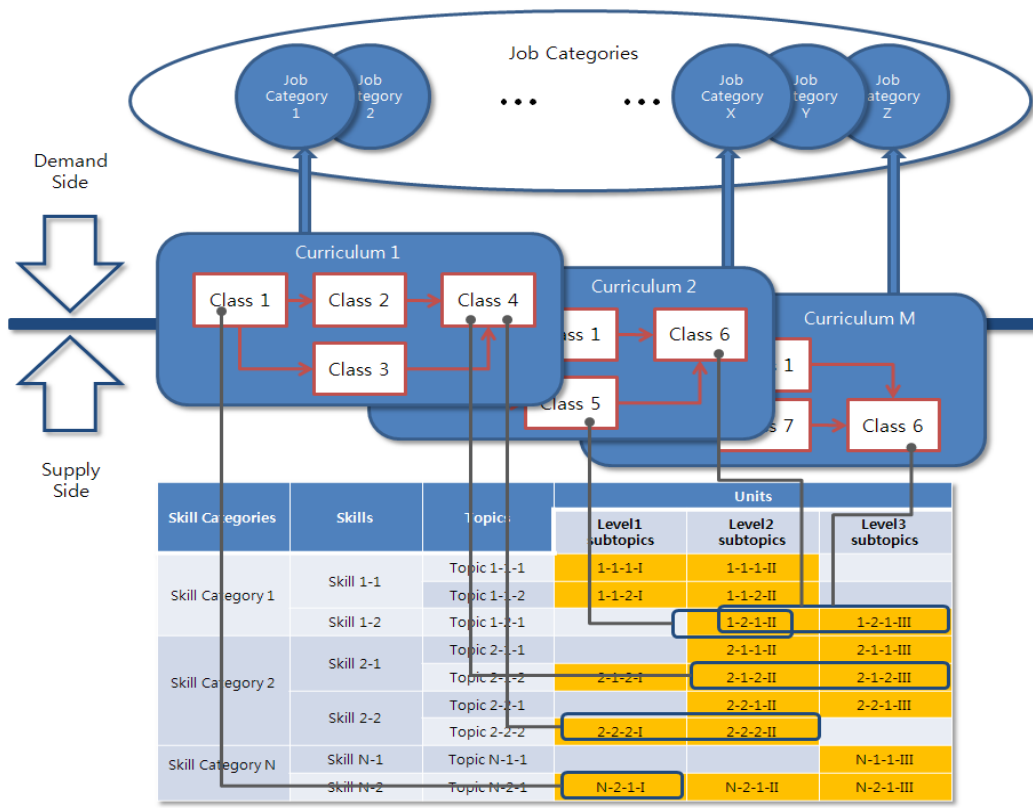


Figure 2. Subtopic-based relationships among skills, classes, courses, and job categories

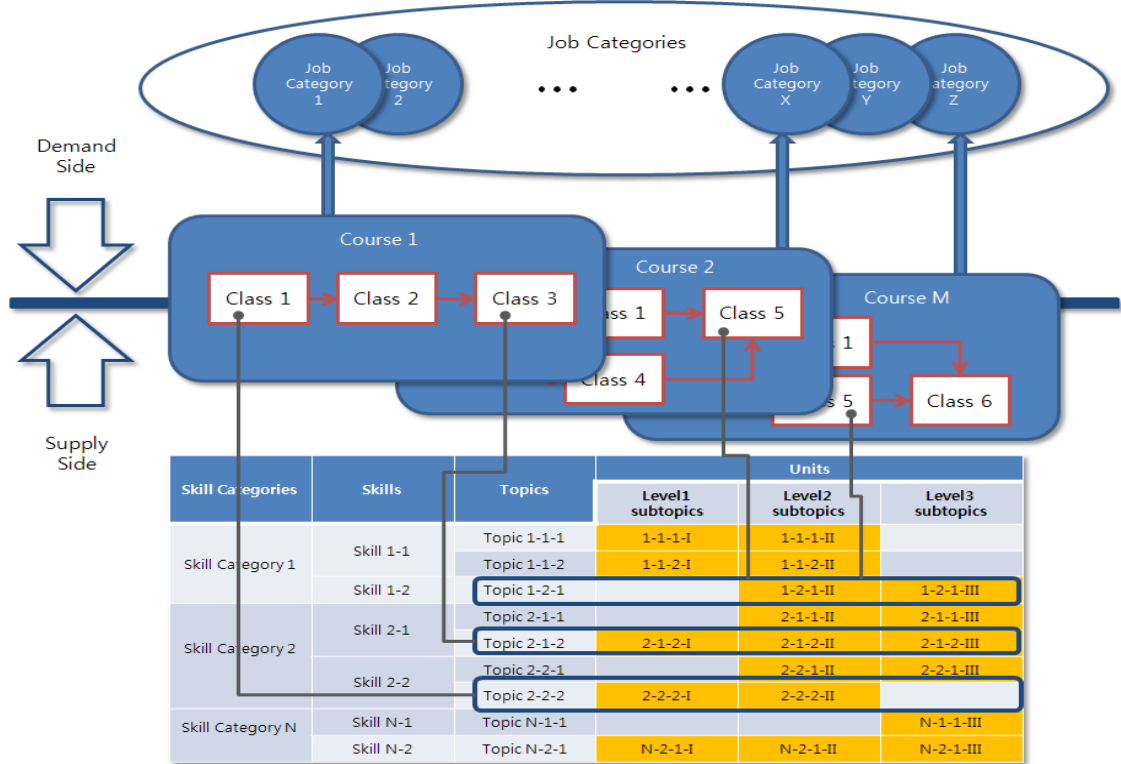


Figure 3. Topic-based relationships among skills, classes, courses, and job categories

1.4 Industrial Masking Sheet & Personal Record Sheet

In the course of employment interviews, a company may need indicators to assess the maturity of a prospective employee. In this document, the industrial masking sheet is introduced as an OSS skill maturity assessment tool. As shown in Figure 4, an industrial masking sheet, that is, “masking sheet” in short, is prepared by a group of companies or a single company in a job category in order to represent the skill specifications for the human resources whom they are seeking. The masking sheet can also be provided by the OSS promotion forum as a recommended industrial masking sheet. The typical form of the masking sheet contains the list of unit codes with check boxes. The checked units are meant to be one of the necessary elementary skills for the specific job.

In contrast, the prospective employee prepares his personal record sheet whose form is similar to the masking sheet. The personal record sheet is typically issued from education and training bodies, or can be issued from authorized certificate agencies which administer the testing procedures. An attempt is made to match the personal record sheet and industrial masking sheet in the course of the employment interview. In some cases, the prospective employee finds some units are missed as shown in Figure 5.

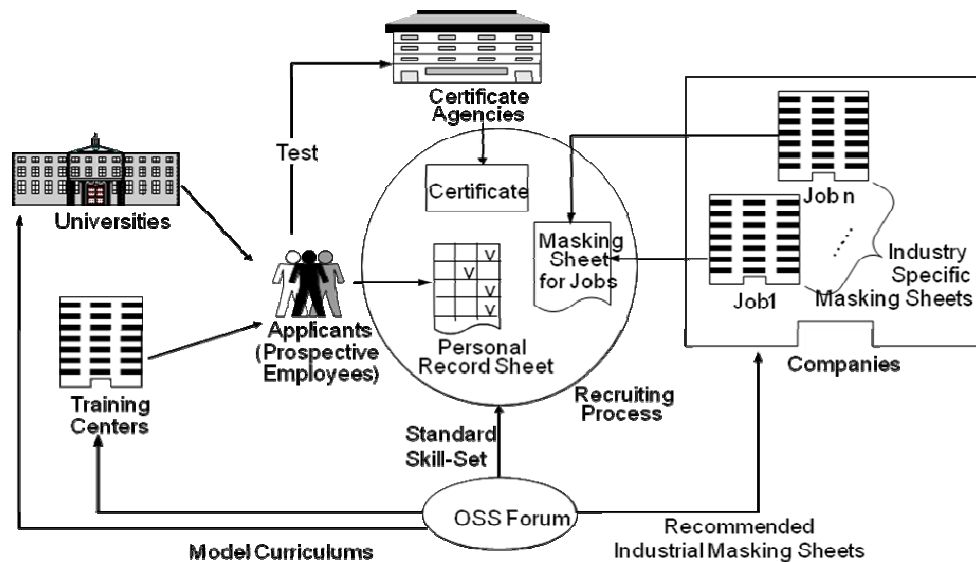


Figure 4. Usage flow of industrial masking sheet and personal record sheet

Personal record sheet from prospective employee

Category(9)	Skills(37)	Topic Check Boxes
Basic	Knowledge of OSS	(V) 1-1-1-I () 1-1-1-II
		(V) 1-1-2-II (V) 1-1-2-III
		() 1-1-3-II
		...



Category(9)	Skills(37)	Topic Check Boxes
Basic	Knowledge of OSS	(V) 1-1-1-I () 1-1-1-II
		(V) 1-1-2-II (V) 1-1-2-III
		(V) 1-1-3-II
		...

Industrial Masking Sheet presented by a company

Figure 5. Example of matching procedure between Industrial Masking Sheet and Personal Record Sheet

2. Detailed Description of Skill-sets

This section provides detailed descriptions of skill-sets that are defined for sample curricula for embedded-application developers. The curricula include nine categories and 38 skills as shown in Table 1. Shaded skills in the table are covered in this section, and the others are in Appendix B.

Table 1 Category of OSS knowledge and skill-sets

Category	WG2 OSS Skills	Code
Basic	Knowledge of OSS	1-1
	Legal Affairs	1-2
	Computer System and Architecture	1-3
	Distributed Architecture	1-4
System	Concept of Linux and Basic Operations	2-1
	Kernel of Linux	2-2
	Linux System Management	2-3
	Linux System Programming	2-4
	Network Server Management	2-5
	Cluster System Architecture	2-6
	Concurrent System Programming	2-7
	Java EE Application Server	2-8
Network	Network Architecture	3-1
	Network Management	3-2
	Network Programming	3-3
Programming	Java	4-1
	C	4-2
	C++	4-3
	Script Language	4-4
	GUI	4-5
	Web Programming	4-6
Multimedia System	Multimedia Service Platform	5-1
	Multimedia Programming	5-2
Development System	Development Frameworks	6-1
	Development Tools	6-2
	Integrated Development Environment	6-3
	Software Testing	6-4
Security	Fundamental of Cryptography	7-1
	Network Security	7-2
	OS Security	7-3
RDB	Basic Skills in RDB	8-1
	RDB System Management	8-2
	RDB Applications Development	8-3
Embedded Software	Embedded System Development	9-1
	Embedded Development Environment	9-2
	Embedded Application Development	9-3
	Embedded System Optimization	9-4
	Device Driver Development	9-5

2.1 Knowledge of OSS

SKILL CATEGORY NAME	Basic Knowledge		SKILL CATEGORY NO.	1	
SKILL NAME	Knowledge of OSS		SKILL NO.	1	
TOPICS	LEVEL	SUBTOPICS		CODE NO.	
Introduction to OSS	I	Objective	Understand the definition of open source software (OSS) and the concept of OSS, OSS development models.	1-1-1-I	
		Prerequisite			
		<ul style="list-style-type: none"> - Understanding the definitions of Open Source Software (OSS) - Understanding the concepts of OSS - Understanding the reasons for business focus on OSS - Understanding OSS development models - Understanding introduction to basic OSS licenses 			
	II	Objective	Understanding situations on deployment and development of OSS		1-1-1-II
		Prerequisite			
		<ul style="list-style-type: none"> - Understanding OSS business models - Understanding types of OSS communities - Understanding the difference between Freeware and OSS 			
History of UNIX and Linux	I	Objective	Describe the history of UNIX from its start in the late 1960s, development into BSD, and contribution to the appearance and development of Linux. In connection with these topics, also address GNU projects and the Free Software Foundation (FSF).	1-1-2-I	
		Prerequisite			
		<ul style="list-style-type: none"> - Understanding the history of UNIX - Understanding the history of Linux 			

		- Understanding the history of GNU Projects		
OSS servers	I	Objective	Understanding that OSS is used most widely in Internet servers (mail, Web, and DNS) and a variety of network servers (file sharing and application servers). Also being aware that OSS middleware is gaining in popularity.	1-1-3-I
		Prerequisite		
		<ul style="list-style-type: none"> - Knowing OSS server software <ul style="list-style-type: none"> • Mail servers • Web servers • DNS servers • File servers • Application servers - Knowing OSS middleware - Knowing OSS virtualization tools 		
OSS development tools	I	Objective	Introduce C/C++, Java, PHP, Perl, Python, and Ruby as languages for OSS development. Stress that these languages are developed in turn using OSS. Explain OSS development frameworks for PHP, Ruby, and Java. Also give a brief description of Eclipse, NetBeans, and WideStudio as OSS integrated development tools.	1-1-4-I
		Prerequisite		
		<ul style="list-style-type: none"> - Knowing C/C++ compilers - Knowing Java environment - Knowing Java tool chain - Knowing PHP environment - Knowing PHP toolkits - Knowing Perl and CPAN - Knowing Python 		

		<ul style="list-style-type: none"> - Knowing Ruby and Ruby on Rails - Knowing development frameworks for PHP, Ruby, and Java - Knowing integrated development tools <ul style="list-style-type: none"> • Eclipse • NetBeans • WideStudio 	
OSS desktop applications	I	Objective	Introduce window systems (integrated desktop environment), office suites, browsers, mailers, and graphic tools as desktop applications based on OSS.
		Prerequisite	
		<ul style="list-style-type: none"> - Knowing window systems <ul style="list-style-type: none"> • X window systems • Window Manager • 3D effected window - Knowing integrated desktop environment <ul style="list-style-type: none"> • KDE • GNOME • XFCE4 - Knowing office suites <ul style="list-style-type: none"> • OpenOffice • KOffice - Knowing browsers <ul style="list-style-type: none"> • Mozilla Firefox • Epiphany • Galeon - Knowing mail clients and schedulers <ul style="list-style-type: none"> • Mozilla Thunderbird • Evolution 	
	1-1-5-I		

			<ul style="list-style-type: none"> • Sylpheed -Knowing graphic tools • GIMP 	
OSS standardization	II	Objective	Outline the trend of OSS standardization. Introduce Linux Standard Base (LSB), which is a standard for Linux, and Java standard specifications to clarify the significance of such standardization. Also address efforts in Asian regions toward standardization and relations with international standardization organizations.	1-1-6-II
		Prerequisite		
		<ul style="list-style-type: none"> - Understanding trends of OSS standardization - Understanding Linux Standard Base (LSB) - Understanding Java standard specifications - Understanding NEAOSS Standardization 		
OSS server applications	I	Objective	Introduce OSS-based server applications – Customer Relation Management (CRM), Business Integrated Data Analysis (BI), Enterprise Resource Planning (ERP), and Content Management System (CMS) for Websites. Also introduce other OSS applications for specific business tasks.	1-1-7-I
		Prerequisite		
		<ul style="list-style-type: none"> - Knowing ERP application <ul style="list-style-type: none"> • OpenERP - Knowing CMS application <ul style="list-style-type: none"> • Zope • Drupal • Plone - Knowing CRM application <ul style="list-style-type: none"> • SugarCRM 		

			- Knowing BI application	
Use of OSS	I	Objective	Outline the current use of OSS by businesses, using data from various studies. Describe the market shares of Linux and OSS middleware. Also, introduce the opinions of business users concerning their interest in adopting OSS and the advantages and disadvantages they perceive in OSS.	1-1-8-I
		Prerequisite		
			<ul style="list-style-type: none"> - Understanding current use of OSS by businesses <ul style="list-style-type: none"> • The market shares of Linux and OSS middleware • The advantages and disadvantages - Understanding Web system construction <ul style="list-style-type: none"> • How OSS is used and points to be clarified in system construction • Pros-and-cons of OSS • Precautions to be taken in system construction - Understanding the OSS community - Understanding and outlining the types and features of OSS communities 	
Web system development	I	Objective	Using a typical example of Web system construction, explain how OSS is used, and clarify points in system construction, pros-and-cons of OSS, and precautions to be taken in system construction.	1-1-9-I
		Prerequisite		
			<ul style="list-style-type: none"> - Understanding typical Web system stacks - Understanding pros and cons of OSS on Web system construction 	
OSS communities	I	Objective	Giving a typical example of an OSS community, outline the types and features of OSS communities. Explain how to join such a community and what precautions to take when participating in one.	1-1-10-I
		Prerequisite		
			<ul style="list-style-type: none"> - Understanding types and features of OSS communities and participation in them 	

		<ul style="list-style-type: none"> - Understanding communities' enabler services - Searching development project - Understanding how to join a project 	
OSS business	II	Objective	
		Prerequisite	
		<ul style="list-style-type: none"> - Understanding business model - Package business - Customizing OSS - Dual licensing - Subscription models - Installation service - Professional services - Support business - Consulting business - Education business 	1-1-10-II
Earning OSS information	II	Objective	
		Prerequisite	
		<ul style="list-style-type: none"> - Project site - SourceForge - Freshmeat - Information sites provided by non profit organizations - News site - Community site - Users' group 	1-1-11-II

Deployment of OSS Operating system	II	Objective	Understanding basics of OSS operating system.	1-1-12-II
		Prerequisite		
		<ul style="list-style-type: none"> - Supported hardware and drivers - OSS Applications - Enterprise Linux distributions <ul style="list-style-type: none"> • SuSE Linux Enterprise Server (SLES) • Red Hat Enterprise Linux (RHEL) - Community based Linux distributions <ul style="list-style-type: none"> • KNOPPIX • Ubuntu - OS other than Linux <ul style="list-style-type: none"> • FreeBSD • NetBSD 		
Deploying server application	II	Objective	Understanding the existence of several kinds of OSS server applications.	1-1-13-II
		Prerequisite		
		<ul style="list-style-type: none"> - MTA <ul style="list-style-type: none"> • sendmail • Postfix - HTTP Server <ul style="list-style-type: none"> • Apache HTTP Server - Installation from source code <ul style="list-style-type: none"> • Configure and create - Installation by binary package <ul style="list-style-type: none"> • Debian package • RPM package form 		
Deployment of OSS	II	Objective	Understanding how to install several OSS server applications.	1-1-14-II

server		Prerequisite	<ul style="list-style-type: none"> - Installation of database management server <ul style="list-style-type: none"> • PostgreSQL • MySQL - Details of MySQL installation <ul style="list-style-type: none"> • Installation of business edition • Installation of community edition - Installation of Network and Server management software <ul style="list-style-type: none"> • OpenNMS • Hinemos - Installation of backup server - Bacula 	
Deployment of OSS desktop application	II	Objective	Understanding how to install and configure several OSS desktop applications.	1-1-15-II
		Prerequisite	<ul style="list-style-type: none"> - Browser <ul style="list-style-type: none"> • Mozilla Firefox and its configuration • Extensions - Mail client <ul style="list-style-type: none"> • Mozilla Thunderbird and its configuration • Connection to POP server • Connection to IMAP4 server - Mail and schedule client <ul style="list-style-type: none"> • Evolution and its configuration • Connection to MS Exchange Server • Connection to IMAP server • Connection to POP3 server 	

		<ul style="list-style-type: none"> - Office applications <ul style="list-style-type: none"> • OpenOffice • Gimp - Open Clip Art 	
Deployment of OSS server application	II	Objective	
		Prerequisite	
		<ul style="list-style-type: none"> - CMS/blogs applications <ul style="list-style-type: none"> • XOOPS cube • Drupal • WordPress - SNS applications <ul style="list-style-type: none"> • OpenPNE • SKIP 	1-1-16-II
Deployment of virtualization software	II	Objective	
		Prerequisite	
		<ul style="list-style-type: none"> - Introduction to virtualization <ul style="list-style-type: none"> • Host OS virtualization • Hypervisor virtualization - Concept of Xen <ul style="list-style-type: none"> • Domain 0 • Domain U - Installation of Xen <ul style="list-style-type: none"> • Installation of guest OS - Configuration of Xen 	1-1-17-II

2.2 Kernel of Linux

SKILL CATEGORY NAME	System		SKILL CATEGORY NO.	2
SKILL NAME	Kernel of Linux		SKILL NO.	2
TOPICS	LEVEL	SUBTOPICS		CODE NO.
General introduction to the kernel of Linux	I	Objective	Understanding the basic roles of OS and the history of OSes which have been developed to date. Knowing today's most popular OSes. Understanding the features of each OS.	2-2-1-I
		Prerequisite	None	
		<ul style="list-style-type: none"> - Understanding the basic roles of OSes - Understanding the background to the deployment of an OS on a computer - Understanding the history of OSes - Understanding the introduction to today's popular OSes - Understanding the type and features of OSes - Understanding the basics of system call - Understanding the basics of context switch 		
Architecture of Linux kernel	I	Objective	Understanding the basic architecture and functions of the Linux kernel, including process management, memory management, file systems, networking, and I/O processing.	2-2-2-I
		Prerequisite	None	
		<ul style="list-style-type: none"> - The architectural view of Linux kernel - Understanding basics of process management - Understanding basics of memory management - Understanding basics of file systems 		

			<ul style="list-style-type: none"> - Understanding basics of networking - Understanding basics of I/O processing 	
Scheduling	I	Objective	Understanding the concept of context switch, and outlining the mechanism of switching processes. Knowing the actual application of scheduling algorithms, in addition to the underlying queuing theory and Markov chain.	2-2-3-I
		Prerequisite		
			<ul style="list-style-type: none"> - Understanding the concept of context switch - Understanding the mechanism of switching processes - Understanding the concept of scheduling algorithm <ul style="list-style-type: none"> • Task list queuing • Real time scheduling • Dual running queue - Using I/O scheduling <ul style="list-style-type: none"> • CFQ • Dead timeout • Elevator algorithm - Understanding the selection and difference of kernel timers <ul style="list-style-type: none"> • TSC • ACPI timer • High resolution timer, HPET 	
Interrupts and delayed action	I	Objective	Understanding the association between the CPU and other devices and the concept of “interrupt.” Knowing several topics on interrupt, including those of the management and types of interrupts. Understanding the outline of interrupt processing in the kernel and the basic idea of the time-sharing system.	2-2-4-I
		Prerequisite		
			<ul style="list-style-type: none"> - Understanding the basics of interrupts 	

		<ul style="list-style-type: none"> • Hardware interrupt • Software interrupt • Timer interrupt <ul style="list-style-type: none"> - Understanding the concept of delayed execution <ul style="list-style-type: none"> • Work queue • Wait queue 		
System calls	I	Objective	Understanding the system call function used to use OS functions in ordinary applications. Understanding the position of the system call among other functions. Knowing the outline of the internal behavior of the OS when the system call function is activated.	2-2-5-I
		Prerequisite		
		<ul style="list-style-type: none"> - Understanding concept of system call - Understanding the trap mechanism - Understanding switch of address space <ul style="list-style-type: none"> • Kernel space • User space • Copying data from/to kernel and user space - Understanding the basic method for implementing features of system call <ul style="list-style-type: none"> • Dispatch table 		
Process management	I	Objective	Understanding the lifecycle of a process as it is created, is completed, and then disappears. Introduce a series of topics related to the lifecycle of processes, including the internal data structure of a process, state transition, process groups, and process-creation methods.	2-2-6-I
		Prerequisite		
		<ul style="list-style-type: none"> - Understanding the concept of processes in multitask OS - Understanding process lifecycle and its process - Understanding the concept of process space 		

		<ul style="list-style-type: none"> - Understanding the transition of running modes - Understanding the concept of process context and kernel context - Understanding the creation of a process and thread <ul style="list-style-type: none"> • fork()/clone() • wait() 		
	II	Objective	Understanding deadlock and other situations where processes need synchronization or exclusion, as well as management by the kernel of such situations. Knowing mounting technology to realize exclusive control, in addition to basic theories underlying the technologies.	2-2-6-II
		Prerequisite		
			<ul style="list-style-type: none"> - Understanding the synchronization and exclusive control - Understanding bad situations such as deadlocks - Understanding the control method of deadlocks in kernel - Understanding mounting technology for exclusive control 	
Memory management	I	Objective	Understanding the concept of “process space,” in which a program runs, and the purpose and features of process spaces. Knowing what can be done in one space but cannot be done in another. Understanding the transition of running modes caused by a system call.	2-2-7-I
		Prerequisite		
			<ul style="list-style-type: none"> - Understanding the principle of physical address management <ul style="list-style-type: none"> • Page • Buddy system • Grab and release of page - Understanding the algorithm of dynamic page allocation <ul style="list-style-type: none"> • Slab allocator • Memory pool 	

		<ul style="list-style-type: none"> • Algorithms for assigning memory <ul style="list-style-type: none"> - Understanding the concept of swap and paging <ul style="list-style-type: none"> • Demand paging • Page cache • Page fault • Dirty page 		
File Systems	I	Objective	Understanding the concept of the virtual file system layer which is an abstract layer for hiding differences in the file system in order to access seamlessly. Understanding the file operation in the kernel and the concept of special files.	2-2-8-I
		Prerequisite		
		<ul style="list-style-type: none"> - Understanding virtual file systems <ul style="list-style-type: none"> • Inode • Directory • Mount - Understanding file operation <ul style="list-style-type: none"> • Open/close • Seek • Read/write - Understanding special files <ul style="list-style-type: none"> • Device file • Proc file system • Sys file system 		
	II	Objective	Understanding the concept of asynchronous access to files and devices to archive high performance.	2-2-8-II
		Prerequisite		
		<ul style="list-style-type: none"> - Understanding asynchronous file operations <ul style="list-style-type: none"> • AIO subsystem 		

		<ul style="list-style-type: none"> - Understanding the concept of direct device access <ul style="list-style-type: none"> • Direct IO 		
Inter process communications	I	Objective	Understanding shared memory, semaphore, messaging, and other functions provided as a means of communicating data and messages between processes.	2-2-9-I
		Prerequisite		
		<ul style="list-style-type: none"> - Understanding IPC Interprocess communication <ul style="list-style-type: none"> • Shared memory • Semaphore • Messaging 		
Protocol stacks		Objective	Understanding the concept of OSI reference models and TCP/IP protocol stacks. Knowing TCP/IP as an example of practical deployed protocol. Understanding the concept of IP address, TCP port number and other advanced features provided by TCP/IP protocol stack.	2-2-10-I
		Prerequisite		
		<ul style="list-style-type: none"> - Understanding socket interfaces <ul style="list-style-type: none"> • UNIX domain socket • INET socket - Understanding IP and UDP <ul style="list-style-type: none"> • IP address • IPv4 and IPv6 • Routing - Understanding UDP and TCP - Difference between two major protocols - TCP flow control and congestion control 		

2.3 Linux System Programming

SKILL CATEGORY NAME	System		SKILL CATEGORY NO.	2
SKILL NAME	Linux System Programming		SKILL NO.	4
TOPICS	LEVEL	SUBTOPICS		CODE NO.
Architecture of Linux Kernel	I	Objective	Understanding the basic architecture and functions of the Linux kernel, including process management, memory management, file systems, networking, and I/O processing.	CODE SHARE 2-2-2-I
		Prerequisite		
System Calls	I	Objective	Understanding the system call function used for OS functions in ordinary applications. Understanding the position of the system call among other functions. Knowing the outline of the internal behavior of the OS when the system call function is activated.	CODE SHARE 2-2-5-I
		Prerequisite		
Linux System Monitoring and Process Management	I	Objective	Understanding the system monitoring tools, including <i>/proc</i> file system and <i>top</i> , <i>ps</i> commands.	2-4-3-I
		Prerequisite	2-3-?-I	
		<ul style="list-style-type: none"> - Understanding how to monitor various parts of the Linux kernel and status of system resources by using /proc file system. - Knowing top, ps commands <ul style="list-style-type: none"> • Command line options • Meaning of the output - Knowing the kill command - Knowing the PowerTOP utility <ul style="list-style-type: none"> • Understanding power-aware system monitoring 		
Development Tools	II	Objective	Knowing how to use GNU tools, including compiler, loader, and debugger. Understanding and writing Makefile to develop Linux system programs.	2-4-4-II

		Prerequisite	6-2-*-I	
			<ul style="list-style-type: none"> - Make and Makefile - Compiler / linker (gcc, GNU ld) - Library management tools - Debugging <ul style="list-style-type: none"> • Debugger • Debugging skill • Strace () 	
Shell script programming	II	Objective	Understanding how to obtain the shell environment and to interface with shell command line arguments.	2-4-5-II
		Prerequisite	2-3-?-I (shell script)	
			<ul style="list-style-type: none"> - Understanding Shell Scripts - Get / put information between C program and shell environments <ul style="list-style-type: none"> • Accessing environment variables • Using command line arguments (argc, argv) - Use redirection and pipe 	
File input/output	I	Objective	Knowing how to use files and directories by using standard POSIX API functions.	2-4-6-I
		Prerequisite		
			<ul style="list-style-type: none"> - POSIX file input/output <ul style="list-style-type: none"> • open(), read(), write(), close(), lseek(), ... • and their stream versions fopen(), ... - Polling functions <ul style="list-style-type: none"> • poll(), select(), epoll() - Mapping file into memory <ul style="list-style-type: none"> • Mmap() 	
	II	Objective	Understanding file control commands to control file and	2-4-6-II

			operations on files, including file locking.	
		Prerequisite		
			<ul style="list-style-type: none"> - Understanding and knowing to use fcntl() commands. <ul style="list-style-type: none"> • F_DUPFD, F_SETFD/F_GETFD, • F_GETFL,/F_SETFL, F_GETOWN/F_SETOWN - Knowing how to use file locking mechanism with fcntl() <ul style="list-style-type: none"> • F_RDLCK, F_WRLCK, F_UNLCK - Understanding blocking, non-blocking operation 	
File systems	II	Objective	Knowing to manage files and directories and device-special files	2-4-7-II
		Prerequisite	2-2-8-I (File System)	
			<ul style="list-style-type: none"> - file and directory manipulation <ul style="list-style-type: none"> • creating, deleting files, directories • link, symbolic link, stat() • accessibility and mktmp, and security - accessing device special files 	
Building a shared library	I	Objective	Understanding the concept of library, and knowing how to manage shared library code.	2-4-8-I
		Prerequisite		
			<ul style="list-style-type: none"> - Understanding the concept of library and using library options in GNU loader <ul style="list-style-type: none"> • static library • shared dynamic library - Understanding the operation library manager <ul style="list-style-type: none"> • The ar command • System library management 	
	II	Objective	Knowing how to build a shared library	2-4-8-II
		Prerequisite		

		- Building a Shared library		
Processes and threads	II	Objective	Knowing how to create process, to execute a program and to manage the process	2-4-9-II
		Prerequisite	2-2-6-I (Process management)	
		<ul style="list-style-type: none"> - Understanding the difference between processes and threads in Linux - Setting the schedule class of a process <ul style="list-style-type: none"> • nice(), sched_getparam(), sched_setparam() - Create process <ul style="list-style-type: none"> • fork() - Execute a program <ul style="list-style-type: none"> • exec() and its variants • system() - Other process-related functions <ul style="list-style-type: none"> • wait(), kill(), ... - Daemonize a process 		
Multi-thread Programming	II	Objective	Knowing how to create, cancel, synchronize with pthread library	2-4-10-II
		Prerequisite	2-2-6-I (Process management)	
		<ul style="list-style-type: none"> - Pthread basic - Thread create, cancellation <ul style="list-style-type: none"> • Passing parameter to thread - Thread synchronization 		
Signals	II	Objective	Knowing how signal works in Linux system, how to catch or ignore signals. Understanding and using setjmp() and longjmp()	2-4-11-II
		Prerequisite		
		<ul style="list-style-type: none"> - Introduction to signal <ul style="list-style-type: none"> • How it works 		

		<ul style="list-style-type: none"> • Side effects of signal <ul style="list-style-type: none"> - Using signal <ul style="list-style-type: none"> • How to catch, ignore signal • Signal handler programming - Understanding and using setjmp(), longjmp() 		
Time	II	Objective	Knowing various time manipulating functions in Linux and knowing how to delay management short or long delay.	2-4-12-II
		Prerequisite	2-1-?-I (Time in Linux)	
		<ul style="list-style-type: none"> - Understating of the presentation of time in Linux - Knowing time-manipulating functions <ul style="list-style-type: none"> • Finding time • Conversion between time formats - Knowing how to wait for time <ul style="list-style-type: none"> • use sleep(), nanosleep(), clocknanosleep() and alarm() • using select() to wait 		
Input/output to/from terminal equipment	I	Objective	Understand tty architecture of Linux and knowing how to use termcap and/or curses library.	2-4-13-I
		Prerequisite		
		<ul style="list-style-type: none"> - Introduction to tty <ul style="list-style-type: none"> • Tty architecture • Tty operation modes • Pseudo tty • Termcap and curses - Understand tty attributes 		
	II	Objective	Knowing how to use termios and curses library	2-4-13-II
		Prerequisite		
IPC mechanisms	II	Objective	Knowing how to use Linux IPC primitives.	2-4-14-II
		Prerequisite	2-2-9-I (Inter-Process Communication)	

		<ul style="list-style-type: none"> - Understanding pipe and named pipe. <ul style="list-style-type: none"> • Using pipe() • Using mkfifo command 		
Network socket programming	II	Objective	Knowing BSD socket programming	2-4-15-II
		Prerequisite	2-2-10-I (Protocol Stacks)	
		<ul style="list-style-type: none"> - Knowing the various socket types and protocols inside <ul style="list-style-type: none"> • UNIX socket, STREAM, DATAGRAM, RAW - Knowing how to TCP/IP client, server application using socket library <ul style="list-style-type: none"> • socket(), close(), bind(), listen(), connect(), accept(), select(), send(), recv(), select(), ... 		
Optimization of Linux system program	III	Objective	Knowing how to measure the performance of a program by profiling and resource usage monitoring.	2-4-16-III
		Prerequisite		
		<ul style="list-style-type: none"> - Knowing to profile application and system <ul style="list-style-type: none"> • Using GNU gprof and gprof library • Oprofile - Optimization techniques 		

2.4 Network Architecture

SKILL CATEGORY NAME	Network		SKILL CATEGORY NO.	3
SKILL NAME	Network Architecture		SKILL NO.	1
TOPICS	LEVEL	SUBTOPICS		CODE NO.
The concept and mechanism of open networks	I	Objective	Understanding basic concept of networking	3-1-1-I
		Prerequisite		
		<ul style="list-style-type: none"> - Networking introduction, characteristics and types <ul style="list-style-type: none"> • Introduction to networking • Packet & circuit switching networks • Connection-oriented and connectionless protocols - Type and size of networks <ul style="list-style-type: none"> • Internet, intranets, and extranets 		
Communications forms and protocols	I	Objective	Understanding network protocol primitives and components	3-1-2-I
		Prerequisite		
		<ul style="list-style-type: none"> - Messages <ul style="list-style-type: none"> • packets, frames, datagrams, and cells • Message formatting - Network protocols - Network performance issue and measurement <ul style="list-style-type: none"> • Performance issue • Understanding performance measurements and units 		
Network communications specifications	I	Objective	Identify standardization of network protocols	3-1-3-I
		Prerequisite		
		<ul style="list-style-type: none"> - Network standards and standards organizations <ul style="list-style-type: none"> • Proprietary, open, de facto standards 		

		<ul style="list-style-type: none"> • Network standards • Internet standards organizations • RFC (Request For Comment) 		
The mechanism of Internet communications	I	Objective	Understand background history behind current Internet and communication models. Identify and describe the roles and capabilities of each seven layers of OSI model, thru practical examples.	3-1-4-I
		Prerequisite		
		<ul style="list-style-type: none"> - Networking fundamentals <ul style="list-style-type: none"> • History of internet communications • The concepts of internet - Open System Interconnection(OSI) reference model <ul style="list-style-type: none"> • OSI model issue and concepts • Physical layer (Layer 1) • Datalink layer (Layer 2) <ul style="list-style-type: none"> ➤ Simplex, full-duplex and half-duplex ➤ Encoding (NRZ, NRZI, Manchester, 4B/5B) • Network layer (Layer 3) • Transport layer (Layer 4) • Session layer (Layer 5) • Presentation layer (Layer 6) • Application layer (Layer 7) 		
Linux network system architecture	I	Objective	Deliver a thorough description of modern communications network design principles and their real implementation on Linux system	3-1-5-I
		Prerequisite	3-1-1-I, 3-1-2-I, 3-1-3-I, 3-1-4-I , Linux commands, shell scripts	
		<ul style="list-style-type: none"> - Linux network system architecture: <ul style="list-style-type: none"> • OSI Layer mapping to Linux network system 		

		<ul style="list-style-type: none"> - Linux network commands: <ul style="list-style-type: none"> • netstat / route / ifconfig / ping - Network configuration files under /etc: <ul style="list-style-type: none"> • rc scripts • Super server (xinetd) configuration - Linux network security basics 		
The mechanism of LAN networks	I	Objective	Introducing basics of LAN (Local Area Network) from signaling/cabling method to most commonly used technology, Ethernet.	3-1-6-I
		Prerequisite		
		<ul style="list-style-type: none"> - The characteristics of LAN - Signaling and Medium <ul style="list-style-type: none"> • Baseband and broadband network • Twisted pair, Coaxial cable, Fiber optic cable and Wireless - Multiple Access <ul style="list-style-type: none"> • CSMA/CD • (Token ring, Token bus) - The topology of LAN <ul style="list-style-type: none"> • Star, Bus, Ring - Ethernet - Switching and forwarding <ul style="list-style-type: none"> • Hub, Bridge, and Switch - Cell switching(ATM) 		
The mechanism of IP networks	I	Objective	Understanding fundamentals of IP network protocols and their implementation including mobile IP.	3-1-7-I
		Prerequisite	3-1-6-I	

		<ul style="list-style-type: none"> - Internet Protocol Versions, Concepts and Overview - IP address <ul style="list-style-type: none"> • Classful address • Classless address (CIDR) • IP Subnetting / Supernetting - Internet Protocol Datagram - Host configuration <ul style="list-style-type: none"> • Host configuration concepts • DHCP - IP encapsulation and tunneling - IP support protocols <ul style="list-style-type: none"> • ICMP error messages type and format • ICMP information messages type and format - IP related protocols <ul style="list-style-type: none"> • ARP • Network address translation (NAT) • IP Security (IPSec) protocol - Mobile IP 		
<p style="text-align: center;">The mechanism of routing</p>	<p style="font-size: 24px;">I</p>	Objective	<p>Introducing basic routing concepts and presenting different routing schemes and their algorithms and metrics.</p>	<p>3-1-8-I</p>
		Prerequisite		
		<ul style="list-style-type: none"> - Routing concepts and features - Routing protocol algorithm and metrics <ul style="list-style-type: none"> • Distance vector • Link-state (Shortest-Path first) routing protocol • Hybrid routing protocol 		

		<ul style="list-style-type: none"> - Routing table and default gateway - Routing protocol specifications <ul style="list-style-type: none"> • Routing Information Protocol <ul style="list-style-type: none"> ➤ RIP, RIP-2, RIPng • Open Shortest Path First (OSPF) • Border Gateway Protocol (BGP/BGP-4) 		
The mechanism of TCP	I	Objective	Understanding the mechanisms and characteristics of TCP and comparison between TCP and UDP. Explanation of network reliability and performance of TCP.	3-1-9-I
		Prerequisite	3-1-6-I, 3-1-7-I, 3-1-8-I	
		<ul style="list-style-type: none"> - TCP/IP overview <ul style="list-style-type: none"> • Architecture • Features and characteristics - TCP / UDP protocol syntax <ul style="list-style-type: none"> • Protocol headers • Addressing <ul style="list-style-type: none"> ➤ Ports and Sockets - TCP data transfer and general operation <ul style="list-style-type: none"> • Connection establishment • Management • Termination - Reliability and Flow Control <ul style="list-style-type: none"> • Segmentation • Retransmission - Congestion control <ul style="list-style-type: none"> • Slow start • Congestion avoidance 		
The mechanism of	I	Objective	Understanding the most used TCP application protocols and	3-1-10-I

TCP applications			their characteristics.	
		Prerequisite	3-1-9-I	
			<ul style="list-style-type: none"> - Web communications <ul style="list-style-type: none"> • http, https - Legacy applications <ul style="list-style-type: none"> • FTP, telnet, SMTP - SSH - DNS - VoIP and IP telephony 	
Types of wireless networks and the mechanism of communications	I	Objective	Introducing the fundamentals of wireless protocols.	3-1-11-I
		Prerequisite	3-1-9-I	
			<ul style="list-style-type: none"> - IEEE 802.11 <ul style="list-style-type: none"> • The architecture and service of IEEE 802.11 networks - Type of wireless networks <ul style="list-style-type: none"> • Wireless PAN • Wireless LAN • Wireless MAN • Mobile device networks - Mobile IP 	
New network architecture	I	Objective	Introducing the new generation of network architectures.	3-1-12-I
		Prerequisite		
			<ul style="list-style-type: none"> - BcN (Beyond Core Network) <ul style="list-style-type: none"> • All IP network concepts • IPTV and streaming • Mobile Internet • MPLS 	

		<ul style="list-style-type: none"> - Next generation of Internet protocol <ul style="list-style-type: none"> • IPv6 • Mobile IPv6 • Communication between IPv4-IPv6 		
Encrypted communications architecture with IPSec	I	Objective	Introducing IPSec, knowing its architecture and operation modes, understanding the algorithms and implementation.	CODE SHARE 7-1-9-I
		Prerequisite	3-1	

2.5 C

SKILL CATEGORY NAME	Programming		SKILL CATEGORY NO.	4
SKILL NAME	C		SKILL NO.	2
TOPICS	LEVEL	SUBTOPICS		CODE NO.
The basics of C	I	Objective	Knowing the basic of C and its history.	4-2-1-I
		Prerequisite		
		<ul style="list-style-type: none"> - Introduction history and features of C language - Comparison of C with other languages - Development cases using C - Flow of operations - C-based program development procedure <ul style="list-style-type: none"> • Creation by an editor to compilation and execution of a program • gcc command line basic options <ul style="list-style-type: none"> ➤ -o, -c, -l • Basic concept of optimizations <ul style="list-style-type: none"> ➤ On 		
Fundamental structure of C	I	Objective	Understanding elements of fundamental structure.	4-2-2-I
		Prerequisite	4-2-1-I The basic of C	
		<ul style="list-style-type: none"> - Configuration of programs, and basic grammar - Basic elements of C programs <ul style="list-style-type: none"> • Variables • Data types • Operators • Control flow statements • Branch with condition. 		

		<ul style="list-style-type: none"> • Loop • Directives and preprocessor. • Prototype definitions. • Character string operation • Array • Functions • Pointers • Structures 		
Concept of data structures	I	Objective	Understanding outline, definition, and usage of some data structures.	4-2-3-I
		Prerequisite		
		<ul style="list-style-type: none"> - Linear lists - Trees - Stacks - Queues - Hash - Other elements of an ordinary data structure. 		
Usage of data structures	I	Objective	Understanding usage of data structures.	4-2-4-I
		Prerequisite	4-2-4-I Concept of data structures	
		<ul style="list-style-type: none"> - Relation between memory management and data structure. - Typical algorithms by applying a data structure. 		
Library	I	Objective		4-2-5-I
		Prerequisite	4-2-2-I Fundamental structure of C	
		<ul style="list-style-type: none"> - Understanding the concept of library and using library options in GNU loader <ul style="list-style-type: none"> • static library • shared dynamic library - Understanding the operation library manage RDC <ul style="list-style-type: none"> • The <i>ar</i> command • System library management 		

Portable C programming	III	Objective	Understanding C program's portability, dependency, using of standard, etc...	4-2-7-III
		Prerequisite	4-2-3-II Library	
		<ul style="list-style-type: none"> - Portability - Dependency by library version - Use of standard <ul style="list-style-type: none"> • POSIX and gcc • LSB(Linux Standard Base) <ul style="list-style-type: none"> ➤ Test suites - Dependency by compiler - gcc extension 		
I18N	II	Objective	Understanding concept of I18N and usage of locale, Multibyte character, etc...	4-2-8-II
		Prerequisite	4-2-3-I Library	
		<ul style="list-style-type: none"> - Locale - Character set <ul style="list-style-type: none"> • Unicode, EUC-KR, GB, JIS, etc... - Multibyte character - Wide character - Multibyte support extension of C functions - GNU gettext - Iconv 		
Checks and tests	II	Objective	Understanding checks and tests C program.	4-2-9-II
		Prerequisite	4-2-3-I Library	
		<ul style="list-style-type: none"> - Internal checks of variables. <ul style="list-style-type: none"> • Argument check 		

		<ul style="list-style-type: none">• Null check- Program for test case- Compile with symbol- Debug- Memory check tool and library<ul style="list-style-type: none">• Valgrind- Electric fence	
--	--	---	--

2.7 Script Languages

SKILL CATEGORY NAME	Programming		SKILL CATEGORY NO.	4
SKILL NAME	Script languages		SKILL NO.	4
TOPICS	LEVEL	SUBTOPICS		CODE NO.
History of Script Languages	I	Objective		4-4-1-I
		Prerequisite		
		<ul style="list-style-type: none"> - History of Script Languages - Execute Environment/Programming-style - Features - Popular Script Languages: Perl/ PHP/ Python/ Ruby - Concept of LAMP Stack 		
Understanding Web architecture and Web 2.0 technology	I	Objective	Describe the Web architecture overview and Web2.0 trends. Also explain the social Web and Web2.0 related components.	4-4-2-I
		Prerequisite		
		<ul style="list-style-type: none"> - Web architecture overview <ul style="list-style-type: none"> • Base Technology in Web 1.0 era • Basic Protocol for Web – HTTP/HTTPS, DNS protocol • CGI (Common Gateway Interface) Overview - Markup / Script Language overview for Web programming <ul style="list-style-type: none"> • HTML • XML • CSS • JavaScript - Introduction to Web 2.0: 		

		<ul style="list-style-type: none"> • Web 2.0 trends overview • Social Web • Folksnomy & Tagging • Mashup • RSS and Atom • Google map API (Geospatial Web) • Semantic Web • Blog 	
Fundamental structure of Perl	I	Objective	
		Prerequisite	
		<ul style="list-style-type: none"> - Basic elements of Perl programs <ul style="list-style-type: none"> • History • Basic syntax • Data types • Variables • Expressions and Operators • Control Structures • Functions - Expressions of Perl 	
			4-4-3-I
Fundamental structure of PHP	I	Objective	
		Prerequisite	
		<ul style="list-style-type: none"> - Basic elements of PHP programs <ul style="list-style-type: none"> • History • Basic syntax • Data types • Variables Constants / • Expressions and Operators • Control Structures • Functions • Object-oriented programming 	
			4-4-4-I

		- Programming-style of PHP: Coding in HTML	
Fundamental structure of Python	I	Objective	
		Prerequisite	
		<ul style="list-style-type: none"> - Basic elements of Python programs <ul style="list-style-type: none"> • History • Lexical analysis • Data model • Expressions and Operators • Simple statement & Compound statement • Object-oriented programming - Data model of Python: Sequences 	
Fundamental structure of Ruby	I	Objective	
		Prerequisite	
		<ul style="list-style-type: none"> - Basic elements of Ruby programs <ul style="list-style-type: none"> • History • Lexical structure • Variables and constants • Operator expressions • Control structure 	
Ruby object-oriented programming	II	Objective	
		Prerequisite	
		<ul style="list-style-type: none"> - What is object-oriented programming <ul style="list-style-type: none"> • Concept • Usage • Example 	
Ruby embedded classes	II	Objective	
		Prerequisite	
		<ul style="list-style-type: none"> - Embedded classes - Embedded modules 	

		- Exception		
Concept of Ruby on rails	II	Objective		4-4-9-II
		Prerequisite		
		- Development process procedures using development frameworks <ul style="list-style-type: none"> • What is a framework? - Ruby on Rails (RoR) <ul style="list-style-type: none"> • Concept • MVC architecture • Installation and setting • Basic usage 		
Database application development	II	Objective		4-4-10-II
		Prerequisite		
		- Using Database on Ruby <ul style="list-style-type: none"> • Active Record • O/R Mapping by Active Record • Connection to Database • Database Migrations - Using Database by Ruby on Rails		
Plug-in installation and development	II	Objective		4-4-11-II
		Prerequisite		
		- Introduction of Plug-in <ul style="list-style-type: none"> • What is a Plug-in • Popular Plug-in - Install Plug-in <ul style="list-style-type: none"> • Development new Plug-in 		
Open source system customization	II	Objective		4-4-12-II
		Prerequisite		
		- Open Source Application		

		<ul style="list-style-type: none"> - Radiant: Open source CMS <ul style="list-style-type: none"> • Installation and Setting • Using Radiant - Customization of Radiant <ul style="list-style-type: none"> • Extension • Install and Uninstall 	
Outline of DOM	I	Objective	
		Prerequisite	
		<ul style="list-style-type: none"> - Introduction of DOM (Document Object Model) - Basic operation of DOM - DOM API 	4-4-13-I
Fundamental structure of JavaScript	I	Objective	
		Prerequisite	
		<ul style="list-style-type: none"> - Introduction of JavaScript - Differences between JAVA and JavaScript - Tag, Variable, Function, Objects, Events - Using Objects, Input Forms, Frames - Stylesheet and layers 	4-4-14-I

2.8 Web Programming

SKILL CATEGORY NAME	Programming		SKILL CATEGORY NO.	4
SKILL NAME	Web Programming		SKILL NO.	6
TOPICS	LEVEL	SUBTOPICS		CODE NO.
Understanding Web architecture and Web 2.0 technology	I	Objective	Describe the Web architecture overview and Web 2.0 trends. Also explain the social Web and Web 2.0 related components.	4-6-1-I
		Prerequisite		
		<ul style="list-style-type: none"> - Web architecture overview <ul style="list-style-type: none"> • Base Technology in the Web 1.0 era • Basic Protocol for Web – HTTP/HTTPS, DNS Protocol • CGI (Common Gateway Interface) Overview - Markup Language overview for Web Programming <ul style="list-style-type: none"> • HTML • XML • CSS - Script Language overview for Web Programming <ul style="list-style-type: none"> • Perl • PHP • Python • Ruby • JavaScript - Introduction to Web 2.0: <ul style="list-style-type: none"> • Web 2.0 trends overview • Social Web • Folksnomy & Tagging • Mashup 		

		<ul style="list-style-type: none"> • RSS and Atom • Google map API (Geospatial Web) • Semantic Web • Blog 	
Markup Language for Web Programming	I	Objective	Present the basic Language for Web programming. Describe Web programming using HTML, CSS, JavaScript and XML.
		Prerequisite	
		<ul style="list-style-type: none"> - HTML Programming <ul style="list-style-type: none"> • Basic HTML Structure • Head, Body, Fonts, Text, Lists, Tables • Link, Object, Frame, • Intrinsic events - Making CSS (Cascading Stylesheet) and Adaptation <ul style="list-style-type: none"> • Introduce CSS • Selectors, Cascading, Box Model • Visual Rendering Model, Visual Rendering Model Details, Colors and Backgrounds • Fonts, Text, Lists, Tables - JavaScript Programming <ul style="list-style-type: none"> • Introduce JavaScript • Different of JAVA and JavaScript • Tag, Variable, Function, Objects, Events • Using Objects, Input Forms, Frames • Stylesheet and layers - XML Programming <ul style="list-style-type: none"> • Introduce XML (eXtended Markup Language) • XML Basic – Structure, Elements, Attribute, Processing Instruction, Entity, Comments 	
	4-6-2-I		

		<ul style="list-style-type: none"> • XML & DTD • XSL & XML • XML in Database – XML/IIS, URL Query, Template file, XDR, XPath 		
Web programming with PHP	II	Objective	Describe Web programming using PHP functions, controls and development methodology.	4-6-3-II
		Prerequisite	4-4-3-I, 4-6-1-I	
		<ul style="list-style-type: none"> - Variables and Expressions in PHP - PHP Operators - Conditional Tests and Events in PHP - PHP Flow Control - Functions in PHP - Arrays - Object-Oriented Programming in PHP - Adding and Accessing Dynamic Content - Cookies - Sessions - File and Directory Access Using PHP - String Manipulation and Regular Expressions - Managing Date and Time - PHP Debugging - MySQL with PHP 		
CGI programming with Perl	II	Objective	Describe Web programming using CGI-Perl functions, controls and development methodology.	4-6-4-II
		Prerequisite	4-4-2-I, 4-6-1-I	
		<ul style="list-style-type: none"> - Introduce CGI Programming with Perl 		

		<ul style="list-style-type: none"> - CGI output with Perl <ul style="list-style-type: none"> • Content Type • Location • Status Line - CGI Input Decoding with Perl <ul style="list-style-type: none"> • GET and POST - CGI Library on Perl <ul style="list-style-type: none"> • cgi-lib.pl • CGI.pm - Other Functions <ul style="list-style-type: none"> • Server Push • Cookie 		
Internet programming with Python	II	Objective	Describe Web programming using CGI-Python functions, controls and development methodology.	4-6-5-II
		Prerequisite	4-4-4-I, 4-6-1-I	
		<ul style="list-style-type: none"> - Introduce CGI Programming with Python - Python Internet Modules <ul style="list-style-type: none"> • Web browser, cgi, urllib, httplib • ftplib, poplib, imaplib • nntplib, smtpplib, telnetlib • urlparse • SocketServer • BaseHTTPServer • SimpleHTTPServer • CGIHTTPServer • Cookies • asyncore • data handlers 		

		<ul style="list-style-type: none"> - Web Programming with Perl <ul style="list-style-type: none"> • CGI (common gateway interface), Cookie module • Database Module – MySQLdb • Apache module - mod_python • Another methods – pmz, fastcgi, Webware • Python http-server modules 					
Web programming with Ruby on Rails	II	<table border="1"> <tr> <td>Objective</td> <td>Present the basic concept and feature of Ruby on Rails. Describe Web programming using Rails functions, controls and development methodology.</td> </tr> <tr> <td>Prerequisite</td> <td>4-4-5-I , 4-4-6-II , 4-4-8-II, 4-4-9-II, 4-6-1-I</td> </tr> </table>	Objective	Present the basic concept and feature of Ruby on Rails. Describe Web programming using Rails functions, controls and development methodology.	Prerequisite	4-4-5-I , 4-4-6-II , 4-4-8-II, 4-4-9-II, 4-6-1-I	4-6-6-II
		Objective	Present the basic concept and feature of Ruby on Rails. Describe Web programming using Rails functions, controls and development methodology.				
		Prerequisite	4-4-5-I , 4-4-6-II , 4-4-8-II, 4-4-9-II, 4-6-1-I				
<ul style="list-style-type: none"> - What are Rails? - Ruby for Rails: <ul style="list-style-type: none"> • Rake • Test/unit - Demystifying Rails: <ul style="list-style-type: none"> • The Console • Models, Controllers • Associations • Migrations • Views • Core Ruby extensions - Test-Driving your Rails App: <ul style="list-style-type: none"> • Beyond Scaffolding • Unit and Functional Tests • Integration Tests - Advanced Controllers and Models: <ul style="list-style-type: none"> • Validations • User Authentication 							

		<ul style="list-style-type: none"> • TDD'ing Actions • Evolving Associations - Rails Power Tools: <ul style="list-style-type: none"> • Mock Objects • Named Routes • Advanced Integration Testing • Rails Plug-ins - Ajax on Rails: <ul style="list-style-type: none"> • Helpers and RJS • Ajax with Prototype and Script.aculo.us • Ajaxifying your Rails App - Advanced Ajax on Rails <ul style="list-style-type: none"> • Helpers and RJS • Ajax with Prototype and Script.aculo.us • Ajaxifying your Rails App 		
Java Programming	I	Objective	Describe Web programming using Java functions, controls and development methodology.	4-6-7-I
		Prerequisite	4-6-1-I	
		<ul style="list-style-type: none"> - Introduce Java Programming - Data Type & Operators - Control Statements & Array - Classes - AWT - Event Control - Applet - Thread 		

		<ul style="list-style-type: none"> - Exception Control - Java I/O (java.io) - Network - JDBC - Object Control 		
Web programming with JSP/Servlet	II	Objective	Present the basic concept and feature of JSP/Servlet. Describe Web programming using JSP/Servlet functions, controls and development methodology.	4-6-8-II
		Prerequisite	4-6-1-I, 4-6-7-II	
		<ul style="list-style-type: none"> - Developing a basic Java Servlet - Developing a View Component - Developing a Controller Component - Developing Dynamic Forms - Sharing Application Resources Using the Servlet Context - Designing the Business Tier - Developing Web Applications Using Struts - Developing Web Applications Using Session Management - Using Filters in Web Applications - Integrating Web Applications With Databases - Developing JSP Pages - Developing JSP Pages Using Custom Tags - Developing Web Applications Using Struts Action Forms - Building Reusable Web Presentation Components 		
XML programming	II	Objective	Present the basic concept XML and feature of XML. Also	4-6-9-II

with Java			explain XML programming using Java and Java related components	
		Prerequisite	4-6-1-I, 4-6-7-II	
		<ul style="list-style-type: none"> - XML Namespace - DTD - Schema - XPath Syntax - XSL - XML, SAX, DOM and JAXP - Java Programming with namespace-aware parser using JAXP - Java Programming using SAX API - Java Programming using DOA API 		
Developing Java Web Services	II	Objective	Describe the Java Web services and Web services related protocols, securities and design guidelines.	4-6-10-II
		Prerequisite	4-6-1-I	
		<ul style="list-style-type: none"> - Web Services overview - Introducing Java technology & platform about Web Services - SOAP - SAAJ - WSDL - Service Registry - JAX-RPC - Overview of Web Services Security - Design guideline of Web Services Security 		

Web Application Development with Ajax	II	Objective	Describe the Ajax and development with Ajax for design guidelines and prototypes.	4-6-11-II
		Prerequisite	4-6-1-I	
		<ul style="list-style-type: none"> - Introduction to Ajax: <ul style="list-style-type: none"> • What is Ajax (where to use it, and why does it matter)? • Synchronous and asynchronous interaction • The XmlHttpRequest object • Retrieving data as text and as XML • Using HTTP methods, headers, and parameters • Asynchronous callback handlers - Ajax Design Basics: <ul style="list-style-type: none"> • Retrieving content • Retrieving executable scripts • Retrieving data: text, XML, JSON • Refactoring the XHR object - Prototype: <ul style="list-style-type: none"> • Extending the core language • New coding idioms for JavaScript technology • Ajax Helper classes • HTML Form and DOM helpers 		
Web 2.0 programming with Java	III	Objective	Describe the Web 2.0 programming with Java and its implementation required for security, services, API and code-base management.	4-6-12-III
		Prerequisite	4-6-1-I, 4-6-7-II	
		<ul style="list-style-type: none"> - Component Models: JSF, Tapestry, Atlas - Exposing Server Objects: DWR, SAJAX - Security concerns: <ul style="list-style-type: none"> • Restricting access 		

		<ul style="list-style-type: none"> • Protecting data • Web 2.0 architecture: Services Integration and Mashups <ul style="list-style-type: none"> - Consuming third-party services: <ul style="list-style-type: none"> • RSS • Google maps • Web services - Exposing a third-party API: <ul style="list-style-type: none"> • SOAP • XML-RPC • REST - Code base management: <ul style="list-style-type: none"> • Profiling JavaScript Code • Testing Ajax applications 		
Internet programming with Google's APP Engine	II	Objective	Describe Web programming using Google App Engine functions, controls and development methodology.	4-6-13-II
		Prerequisite	4-6-1-I, 4-6-5-II, 4-6-7-II	
		<ul style="list-style-type: none"> - What is Google App Engine? - Google App Engine SDK for Python <ul style="list-style-type: none"> • Overview • Servlet Environment • Storing Data • Services • Javadoc Reference • JRE Class White List • Configuration • How-To - Google App Engine SDK for Java <ul style="list-style-type: none"> • Overview 		

		<ul style="list-style-type: none">• CGI Environment• Storing Data• Services• Configuration• How-To- Managing Your App<ul style="list-style-type: none">• The Admin Console• Quotas• Billing	
--	--	--	--

2.9 Fundamentals of Cryptography

SKILL CATEGORY NAME	Security		SKILL CATEGORY NO.	7
SKILL NAME	Fundamental of Cryptography		SKILL NO.	1
TOPICS	LEVEL	SUBTOPICS		CODE NO.
Security functions and cryptography positioning	I	Objective	Knowing the basic concepts and the big picture of security, knowing the risks in Open Source Software (OSS).	7-1-1-I
		Prerequisite		
		<ul style="list-style-type: none"> - Introduction of the basic concepts of information security and knowing the risks in Open Source Software. - Knowing the basic knowledge and history of cryptography. - Knowing which objectives of information security can be achieved with cryptography <ul style="list-style-type: none"> • privacy or confidentiality • data integrity • authentication • non-repudiation. - Background on functions - Basic terminology and concepts - Knowing the classification, i.e. symmetric key encryption and asymmetric key encryption. 		
Cryptography systems/common key	II	Objective	Introducing the symmetric key encryption, knowing its basic concepts, features, advantages and disadvantages, understanding	7-1-2-II

cipher systems		Prerequisite	<ul style="list-style-type: none"> - Finite field. - Linear Feedback Shift Register (LFSR) 	
		<ul style="list-style-type: none"> - Introduction - Pseudorandom bits and sequences - Stream ciphers <ul style="list-style-type: none"> • Feedback shift registers • Stream ciphers based on LFSRs • Other stream ciphers - Block ciphers <ul style="list-style-type: none"> • Classical ciphers and historical development • DES and AES • RC5, IDEA and other block ciphers - Some known attacks 		
Cryptography systems/public key cipher systems	II	Objective	Introducing the public key encryption, knowing its basic concepts, features, advantages and disadvantages, understanding its mechanisms.	7-1-3-II
		Prerequisite	<ul style="list-style-type: none"> - Information theory - Complexity theory - Element number theory - Abstract Algebra 	

		<ul style="list-style-type: none"> - Introduction - Goals of Public Key Cryptography - Key Exchange and Digital Signature - RSA public-key cryptosystem - Rabin public-key cryptosystem - ElGamal public-key cryptosystem - Elliptic curve public-key cryptosystem - NTRU public-key cryptosystem - Key Length and Security 		
Cryptography Toolkits	III	Objective	Knowing the typical cryptography toolkits, understanding the usage of them.	7-1-3- III
		Prerequisite		
		<ul style="list-style-type: none"> - Operating System Crypto Services, - Crypto Library of OpenSSL Toolkit - Gnu Libgcrypt - Mozilla NSS Security - Cryptography Hardware Support, PKCS #11 Token, BSD/Solaris /dev/crypto hardware crypto device 		
Cryptography application systems in information systems	II	Objective	Understanding the cryptography application systems in information systems, including the database encryption, file encryption, etc.	7-1-4-II
		Prerequisite		
		<ul style="list-style-type: none"> - Introduction - Database encryption - File encryption 		
Digital signature	II	Objective	Understanding the digital signature mechanisms, understanding the features and usage of typical digital signatures.	7-1-5-II

		Prerequisite		
		<ul style="list-style-type: none"> - Introduction - A framework for digital signature mechanisms - RSA and related signature schemes - Fiat-Shamir signature schemes - The DSA and related signature schemes - One-time digital signatures - Other signature schemes - Signatures with additional functionality - Digital certificate 		
OSS utilization scenes	II	Objective	Knowing the necessity of cryptography in different OSS application scenes. Knowing the implementation of encryption processing with OSS at the OS, middleware, and application levels.	7-1-6-II
		Prerequisite		
		<ul style="list-style-type: none"> - Introduction - Software encryption - Trust computation and software authentication - Web Service Security 		
Cryptography usage in Wireless LAN	II	Objective	Introducing the cryptography usage in WLAN, knowing its necessity, features, advantages and disadvantages. Knowing the implementation of wireless networks encryption.	7-1-7-II
		Prerequisite	3-1	

		<ul style="list-style-type: none"> - Introduction - Attacks in Wireless LAN - Types of unauthorized access in Wireless LAN - 802.11i security - Implementation of Wireless network encryption 		
Authentication	I	Objective	Knowing the basic mechanism and implementation method. Understanding practical methods to achieve a proper authentication system.	7-1-8-I
		Prerequisite		
	<ul style="list-style-type: none"> - Introduction - The basic concept of authentication - Passwords (weak authentication) 			
	II	<ul style="list-style-type: none"> - Challenge-response identification (strong authentication) - Customized and zero-knowledge authentication protocols - Attacks on authentication protocols 		7-1-8-II
Encrypted communications using IPsec	I	Objective	Introducing IPsec, knowing its architecture and operation modes, understanding the algorithms and implementation.	7-1-9- I
		Prerequisite	3-1	
	<ul style="list-style-type: none"> - Introduction - Security architecture of IPsec - Modes of operation 			
	II	<ul style="list-style-type: none"> - Cryptographic algorithms in IPsec - Software implementations - Standards status 		7-1-9- II
SSH	II	Objective	Introducing Secure Shell (SSH), knowing its outline, specifications and features, understanding the usage of SSH.	7-1-10- II

		Prerequisite		
		<ul style="list-style-type: none"> - Introduction - The history - SSH architecture - Usage of SSH <ul style="list-style-type: none"> • tunneling - Security cautions of SSH 		
SSL/TLS	II	Objective	Introducing Secure Socket Layer (SSL), knowing its outline, specifications, and features, understanding related applications	7-1-11- II
		Prerequisite		
		<ul style="list-style-type: none"> - Introduction - SSL/TLS Standard in RFC - Security Features of SSL/TLS - SSL on the Web/ HTTPS - Programming with SSL/OpenSSL - SSL/TLS on mail protocols 		
VPN communications	III	Objective	Understanding the functions and mechanism of VPN.	7-1-12-III
		Prerequisite	7-1-9- II	

		<ul style="list-style-type: none"> - VPN Routing - User-visible PPVPN services <ul style="list-style-type: none"> • Virtual private wire and private line services (VPWS and VPLS) • Virtual private LAN service (VPLS) • IP-only LAN-like service (IPLS) - Categorizing VPN security models - Authentication before VPN connection - Trusted delivery networks - Security mechanisms and mobility - OSS implementations <ul style="list-style-type: none"> • PoPToP • OpenVPN • OpenSWan and StrongSwan 		
<p>The mechanism of PKI (public key encryption infrastructure)</p>	<p>III</p>	<p>Objective</p>	<p>Knowing the purpose, role, and fields of application of Public Key Infrastructure (PKI), understanding the mechanism and the implementation of it.</p>	<p>7-1-13- III</p>
		<p>Prerequisite</p>	<p>7-1-3- I 7-1-5- I</p>	
		<ul style="list-style-type: none"> - The overview of PKI - Basic problems of confidence and trust - The basic model of X.509 PKI, including the Certificate Authority (CA), Registration Authority (RA) and Certificate Distribution System. - The work flow of X.509 PKI: how does PKI work? - Certificate Hierarchies and Cross-Certification. - The revocation mechanism of digital certificate, including Certificate Revocation List (CRL) and Online Certificate Status Protocol (OCSP). - Certificate Operations with OpenSSL and Browsers 		

		- PKI enabled applications.	
PGP	III	Objective	Understanding the mechanism and usage of PGP.
		Prerequisite	
		<ul style="list-style-type: none"> - Introduction - PGP and SMIME - PGP Web of Trust - PGP and GnuPG - File Encryption with PGP/GnuPG - Secure e-mail with PGP/GnuPG - package signing 	
			7-1-14-III

2.10 Network Security

SKILL CATEGORY NAME	Security		SKILL CATEGORY NO.	7
SKILL NAME	Network Security		SKILL NO.	2
TOPICS	LEVEL	SUBTOPICS		CODE NO.
Outline of network security	I	Objective	Knowing The basics of network security.	7-2-1-I
		Prerequisite		
		<ul style="list-style-type: none"> - Introduction to network security - Background and Problem - Network security model - Attack process - Attack technology - Malicious code - Development 		
Virus characteristics and antivirus measures	I	Objective	Understanding virus characteristics and antivirus	7-2-2-I
		Prerequisite	7-2-1-I The basics of network security	
		<ul style="list-style-type: none"> - Virus overview - Virus characteristics - The kinds of viruses <ul style="list-style-type: none"> • Boot virus • File virus • Micro virus • Mutation virus • U disk virus - Network virus 		

		<ul style="list-style-type: none"> • Trojan • Worm • Mobile virus 		
	II	Objective	Understanding antivirus	
		Prerequisite	7-2-I-I Understanding virus characteristics	
		<ul style="list-style-type: none"> - Anti-virus measures <ul style="list-style-type: none"> • Characteristics scan • Behavior audit • Active kernel defense • System patch • Other antivirus measures 		7-2-2-II
Simplified classification of network attack methods	I	Objective	Understanding network attack methods	
		Prerequisite	7-2-1-I The basics of network security	
		<ul style="list-style-type: none"> - Attack methods overview - Attack Technology <ul style="list-style-type: none"> • Scan technology • Password attack • DDOS • Buffer overflow • Google hacking • Network fishing • SQL injection • Convert TCP Attack • etc 		7-2-3-I
Unauthorized access techniques by exploiting TCP	I	Objective	Knowing the techniques of exploiting TCP	
		Prerequisite	7-2-1-I The basics of network security	
		<ul style="list-style-type: none"> - Introduction to TCP - TCP protocol - Vulnerability of TCP 		7-2-4-I

		- Exploiting for TCP	
Attacks on the Web	I	Objective	Knowing the techniques of Attacking on the Web
		Prerequisite	7-2-1-I The basics of network security
		<ul style="list-style-type: none"> - Introduction to Web - Security mechanism of Web <ul style="list-style-type: none"> • IIS • Apache • Tomcat - Vulnerability of Web - Exploiting for Web - Exploitation Code 	
			7-2-5-I
Unauthorized access techniques by exploiting IP	II	Objective	Knowing the techniques of exploiting IP
		Prerequisite	7-2-1-I The basics of network security
		<ul style="list-style-type: none"> - Introduction to IP - IP protocol - Vulnerability of IP - Exploiting for IP 	
			7-2-6-II
TCP/IP network security design methods	II	Objective	Knowing TCP/IP network security design methods
		Prerequisite	7-2-1-I The basics of network security
		<ul style="list-style-type: none"> - Introduction to TCP/IP network security - Risk evaluation - Security engineering - Survivability methods - Security architecture design 	
			7-2-7-II
Mechanism of access control, and firewall functions	II	Objective	Knowing the mechanism of firewall
		Prerequisite	7-2-1-I The basics of network security
		<ul style="list-style-type: none"> - Introduction to access control 	
			7-2-8-II

		<ul style="list-style-type: none"> • DAC/MAC - Firewall overview - Functions <ul style="list-style-type: none"> • Router • Payload distribution • VLAN • NAT • Filter • Monitor • QOS • VPN 		
Network vulnerability investigation	II	Objective	Knowing network vulnerability investigation	7-2-9-II
		Prerequisite	7-2-1-I The basics of network security	
		<ul style="list-style-type: none"> - Network vulnerability overview - TCP/IP protocol security - Security of network device <ul style="list-style-type: none"> • Router vulnerability • Switch vulnerability • IDS • Firewall - Fuzz Testing 		
Intrusion detection system specifications and installation	II	Objective	Knowing network vulnerability investigation	7-2-10-II
		Prerequisite	7-2-1-I The basics of network security	
		<ul style="list-style-type: none"> - Introduction to IDS - System Specifications - IDS Installation <ul style="list-style-type: none"> • RealSecure • Snort 		

		• Session Wall		
Mobile computing and remote access security	II	Objective	Knowing network vulnerability investigation	7-2-11-II
		Prerequisite	7-2-1-I The basics of network security	
		<ul style="list-style-type: none"> - Mobile computing - Remote access - Ad hoc network - Security - Mobile Agent - Authority 		
Linux network security measures	II	Objective	Knowing network vulnerability investigation	7-2-12-II
		Prerequisite	7-2-1-I The basics of network security	
		<ul style="list-style-type: none"> - Network vulnerability overview - TCP/IP protocol security - Security of network device <ul style="list-style-type: none"> • Router vulnerability • Switch vulnerability • IDS • Firewall - Fuzz Testing 		
Network security creation	II	Objective	Knowing network security creation	7-2-13-II
		Prerequisite	7-2-1-I The basics of network security	
		<ul style="list-style-type: none"> - Principle - Method - Architecture - Secure Model - Practice example 		
Secure network creation	II	Objective	Knowing secure network creation	7-2-14-II
		Prerequisite	7-2-1-I The basics of network security	

		<ul style="list-style-type: none"> - Principle - Information collection - Scanning - Assessment - Suggest of secure network 		
Mechanism of intrusion detection systems	II	Objective	Knowing network vulnerability investigation	7-2-15-II
		Prerequisite	7-2-1-I The basics of network security	
		<ul style="list-style-type: none"> - Principle - Attack Model - Function - Detection policy - Countermeasures of IDS 		

2.11 OS Security

SKILL CATEGORY NAME	Security		SKILL CATEGORY NO.	7
SKILL NAME	OS Security		SKILL NO.	3
TOPICS	LEVEL	SUBTOPICS		CODE NO.
OS Security Concepts	I	Objective	Understanding security threats and attacks on operating systems. Understanding basic concepts on operating system security Understanding related criteria on secure operating system Understanding OS security related criteria	7-3-1-I
		Prerequisite	7-2-?-I	
		<ul style="list-style-type: none"> - Understanding OS Security threats and attacks <ul style="list-style-type: none"> • virus & worms • Trojan horse • buffer overflow • logic bomb • covert channel - Understanding basic OS security concepts <ul style="list-style-type: none"> • reference monitor • trusted computing base • security function & security assurance - Understanding OS security related criteria <ul style="list-style-type: none"> • TCSEC • CC • Other criteria 		
OS Security Mechanisms	I	Objective	Understanding basic security mechanisms in trusted operating systems, including identification & authentication, discretionary & mandatory access control, integrity protection, least privilege management, trusted	7-3-2-I

			path, trusted recovery, covert channel, object reuse, audit and so on.	
		Prerequisite	7-3-1-I	
		<ul style="list-style-type: none"> - Understanding basic security mechanisms in trusted operating systems, including: <ul style="list-style-type: none"> • identification & authentication • discretionary & mandatory access control • integrity protection • least privilege management • trusted path • trusted recovery • covert channel • object reuse • audit 		
	II	Objective	Knowing how to use current security mechanisms in Linux systems.	7-3-2-II
		Prerequisite	2-3-?-I	
		<ul style="list-style-type: none"> - Knowing how to use user identification and authentication commands and related files <ul style="list-style-type: none"> • Commands, e.g. <i>useradd</i>, <i>userdel</i>, <i>usermod</i>, <i>password</i> ... • Files, e.g. <i>/etc/passwd</i>, <i>etc/shadow</i>, ... - Knowing how to use discretionary access control commands <ul style="list-style-type: none"> • 9 bit mechanism, e.g. <i>chmod</i>, <i>chown</i>, ... commands • ACL (Access Control List) mechanism, e.g. <i>setfacl</i>, <i>chfacl</i>, <i>getfacl</i>,... commands - Knowing how to use audit and log services and files, e.g. <i>syslogd</i>... , <i>/etc/syslog.conf</i>, ... - Knowing how to use network security related services and files, e.g. <i>/etc/xinetd.d</i>, <i>/etc/services</i> ... , <i>etc/hosts.allow</i>, <i>etc/hosts.deny</i>, ... 		
OS Security Models	II	Objective	Understanding typical and practical security models in Linux systems	7-3-3-II
		Prerequisite	7-3-1-I	
		<ul style="list-style-type: none"> - Understanding multilevel security policy and confidentiality model <ul style="list-style-type: none"> • <i>Bell-LaPadula</i> Model - Understanding integrity protection policy and integrity models <ul style="list-style-type: none"> • <i>Biba</i> 		

		<ul style="list-style-type: none"> • <i>Clark-Wilson</i> <ul style="list-style-type: none"> - Understanding multi-policy security and neutral security models <ul style="list-style-type: none"> • <i>RBAC</i> (Role Based Access Control) • <i>UCON</i> (Usage Control) • <i>TE/DTE</i> (Type Enforcement/Domain and Type Enforcement) • Chinese Wall - Understanding MAC (mandatory access control) - Other security models (selected) <ul style="list-style-type: none"> • Information flow • Non-interference 		
Linux Security Module-LSM	II	Objective	Understanding flask security architecture and LSM framework	7-3-4-II
		Prerequisite	2-2-?-I, 2-4-?-I, 4-2-?-I	
		<ul style="list-style-type: none"> - Knowing the structure of Linux security module and hook functions <ul style="list-style-type: none"> • Opaque security fields such as <i>void *security</i> • Calls to security hook functions • Security system call such as <i>sys_security()</i> • Registering security modules • Capabilities - Understanding what LSM can and cannot do. 		
	III	Objective	Knowing how to design and implement security mechanisms in Linux.	7-3-4-III
		Prerequisite	2-2-?-I, 2-4-?-I, 4-2-?-I	
		<ul style="list-style-type: none"> - Knowing how to implement security mechanisms with LSM framework in Linux <ul style="list-style-type: none"> • DTE Linux <ul style="list-style-type: none"> ➤ DTE Policy ➤ DTE Implementation ➤ How to use • POSIX.1e Capabilities <ul style="list-style-type: none"> ➤ Capability Definition ➤ Capability Implementation 		

		➤ How to use		
SELinux	II	Objective	Understanding SELinux MAC security policies and policy language	7-3-5-II
		Prerequisite	7-3-3-II, 7-3-4-II,7-3-4-III	
		<ul style="list-style-type: none"> - Understanding the Configurable policies in SELinux <ul style="list-style-type: none"> • Type Enforcement (TE) • Role Based Access Control (RBAC) • Multi Level Security (MLS) and Multi Category Security (MCS) - Understanding flask security architecture <ul style="list-style-type: none"> • Object manager • Security server - Understanding what SELinux can and cannot do. - Understanding and knowing how to use SELinux policy language <ul style="list-style-type: none"> • Ability to enforce confidentiality and integrity guarantees. • Ability to confine flawed and malicious applications. 		
	III	Objective	Knowing how to use the policy language, related security management and policy analysis tools. Knowing SELinux Security architecture and implementation.	7-3-5-III
		Prerequisite	7-3-3-II, 7-3-4-II,7-3-4-III, 2-2-?-I, 2-4-?-I, 4-2-?-I	
	II	<ul style="list-style-type: none"> - Knowing the SELinux architecture and implementation <ul style="list-style-type: none"> • Flask architecture SHARED CODE 7-3-3-I • SELinux LSM Module <ul style="list-style-type: none"> ➤ Data structures ➤ Hook Functions ➤ Security system calls ➤ Module registering - Knowing how to use SELinux management and policy analysis tools <ul style="list-style-type: none"> • <i>checkpolicy</i> • <i>libselinux</i> • <i>libsemanage</i> 		

		<ul style="list-style-type: none"> • <i>libsepol</i> • <i>policycoreutils</i> 	
Various implementations	II	Objective	Understanding characteristics of various implementations.
		Prerequisite	
		<ul style="list-style-type: none"> - SMACK - TOMOYO Linux - AppArmor - grsecurity 	
			7-3-6-II

2.12 Basic Skills in RDB

SKILL CATEGORY NAME	RDB		SKILL CATEGORY NO.	8
SKILL NAME	Basic Skills in RDB		SKILL NO.	1
TOPICS	LEVEL	SUBTOPICS		CODE NO.
Basic database theory	I	Objective		8-1-1-I
		Prerequisite		
		<ul style="list-style-type: none"> - The advantages of databases and the status of Database Management Systems (DBMSs) <ul style="list-style-type: none"> • Concurrent access • Integrity - Basic concepts of using databases <ul style="list-style-type: none"> • OSS implementation • Outline of database structure 		
Fundamental knowledge of RDBMSs	I	Objective		8-1-2-I
		Prerequisite		
		<ul style="list-style-type: none"> - Relational model as a data structure model. - Relational Database Management System (RDBMS) based on a relational model. - Outline database language SQL. - DDL - DML 		
Fundamental concepts of transactions	I	Objective		8-1-3-I
		Prerequisite		
		<ul style="list-style-type: none"> - The concept and features of transaction indispensable to database design. <ul style="list-style-type: none"> • ACID • Concurrent Execution • Recovery - Concept and implementation of exclusive control. 		

		<ul style="list-style-type: none"> - Lock <ul style="list-style-type: none"> • Shared lock • Exclusive lock • Deadlock • 2-phase lock - Isolation level <ul style="list-style-type: none"> • Dirty reads • Non-repeatable reads • Phantom reads • READ UNCOMMITTED • READ COMMITTED • REPEATABLE READ • SERIALIZABLE 	
<p style="text-align: center;">Database components</p>	<p>I</p>	Objective	
		Prerequisite	
		<ul style="list-style-type: none"> - Definition of the components of a relational database <ul style="list-style-type: none"> • Table • Attribute • Tuple - Relational schemas <ul style="list-style-type: none"> • Introduction relational schemas • Main keys to identify only one record • Main keys having plural attributes • External keys. - Set operations <ul style="list-style-type: none"> • Union • Difference • Intersection • Cartesian product - Relational algebraic operations 	<p>8-1-4-I</p>

		<ul style="list-style-type: none"> • Projection • Join • Selection (Restriction) • Division <p>– Ways to maintain consistency among components of a relational database, as well as conditions necessary to maintain consistency.</p>	
DOA	I	Objective	
		Prerequisite	
		<p>–Outline of DOA (Data-Oriented Approach)</p> <ul style="list-style-type: none"> • Difference from the conventional process design approaches. <p>–Data models</p> <ul style="list-style-type: none"> • Conceptual data model • Logical data model • Physical data model <p>–Three-layer schemas.</p> <ul style="list-style-type: none"> • Conceptual schema • External schema • Internal schema 	
Basic database design theory	I	Objective	
		Prerequisite	
		<p>– Methods and procedures for database design</p> <ul style="list-style-type: none"> • Business analysis • Conceptual design • Physical design • Database design <p>– Two analytic techniques</p> <ul style="list-style-type: none"> • The top-down approach • The bottom-up approach. 	
ER models	I	Objective	

		Prerequisite	
		<ul style="list-style-type: none"> - Outline of ER model (Entity-Relational model) - Data representation methods <ul style="list-style-type: none"> • Entity • Relationship • Attribute • Cardinality - ERD (Entity Relational Diagram) <ul style="list-style-type: none"> • Components 	
Normalization	I	Objective	
		Prerequisite	
		<ul style="list-style-type: none"> - Outline of normalization <ul style="list-style-type: none"> • Basic concepts • Functional dependency • Types of normal forms. - Normalization procedures and methods 	8-1-8-I
Database indexes	II	Objective	
		Prerequisite	
		<ul style="list-style-type: none"> - Concept of index - Principle of index <ul style="list-style-type: none"> • Algorithms of retrieval - Implementation of index <ul style="list-style-type: none"> • Physical structure • Performance • Recovery characteristics of index. - Features, advantages and disadvantages with various types of index <ul style="list-style-type: none"> • Binary tree (B tree) • Index-organized tables 	8-1-9-II

		<ul style="list-style-type: none"> • Bitmap index • Reverse key index • Hash cluster (hash index) 	
The database physical structure	II	Objective	
		Prerequisite	
		<ul style="list-style-type: none"> - Physical structure of databases required in the implementation of relational databases. - Purpose of physical design of databases, and procedures required in the physical design <ul style="list-style-type: none"> • Tradeoff to note in the design • How to locate data, index and log • Securing database storage area. 	8-1-10-II
Database access in SQL	II	Objective	
		Prerequisite	
		<ul style="list-style-type: none"> - SQL (Structured Query Language) <ul style="list-style-type: none"> • Structure • Principle • Operation specifications • Types • Features. - Creation of tables <ul style="list-style-type: none"> • Data type specification • Handling of NULL • Specifying procedures of primary key and foreign key. - Data retrieval and query processing <ul style="list-style-type: none"> • SELECT statements <ul style="list-style-type: none"> ➤ Processing of data <ul style="list-style-type: none"> ◆ Expressions ◆ Numerical values ◆ Character strings ◆ Dates ➤ Operations using WHERE clauses 	8-1-11-II

		<ul style="list-style-type: none"> ➤ Set functions. - Joins of tables <ul style="list-style-type: none"> • Outer joins • Self joins • Sub query - Various table handling <ul style="list-style-type: none"> • Complicated retrieval/inquiry procedures • Updating/deleting of tables 	
Typical open source RDBMS products	II	Objective	
		Prerequisite	
		<ul style="list-style-type: none"> - Following typical open source RDBMSs's history of developments, main functions, features, and their licenses <ul style="list-style-type: none"> • PostgreSQL • Firebird • MySQL • SQLite - Comparison with commercial RDBMS - Reason of using open source RDBMS - Portability - Difference among RDBMSs <ul style="list-style-type: none"> • Backup (online and offline) • Specialized features <ul style="list-style-type: none"> ➤ Vacuum 	
Database design/creation practice	II	Objective	
		Prerequisite	
		<ul style="list-style-type: none"> - Business analysis <ul style="list-style-type: none"> • Screening out entities • Creation of ERD - Design procedure of databases 	
			8-1-12-II
			8-1-13-II

		<ul style="list-style-type: none">• Schema• Index• Distributed databases- Physical designs- Creation of tables<ul style="list-style-type: none">• Restriction• Data loading- Items required in actual DB application developments<ul style="list-style-type: none">• Procedures for the designs of business forms and screens- Performance<ul style="list-style-type: none">• Data size and query plan• Improvement of index design• Monitoring of performance	
--	--	---	--

2.14 Embedded Systems Development

SKILL CATEGORY NAME	Embedded Software		SKILL CATEGORY NO.	9
SKILL NAME	Embedded Systems Development		SKILL NO.	1
TOPICS	LEVEL	SUBTOPICS		CODE NO.
Basics of embedded computer systems	I	Objective	Understanding the basic concept and substance of embedded computer systems. Describe the general architecture of hardware, software, and networks in embedded computer systems.	9-1-1-I
		Prerequisite		
		<ul style="list-style-type: none"> - Basics of embedded computer system - Application of embedded computer systems - Architectures for embedded computer systems - Software architecture for embedded systems - Network architecture for embedded systems - Design methods for embedded computer systems - Needs and application for OSS embedded computer systems - Relations between embedded systems and OSS - Practical cases of OSS embedded OSS systems - Precautions to be taken in using OSS in embedded systems 		
Architecture of embedded computer systems	I	Objective	Understanding basic structure and roles and characteristics of embedded computer systems	9-1-2-I
		Prerequisite	9-1-1-I	

		<ul style="list-style-type: none"> - Characteristics of embedded hardware - Requirement of embedded computer system - Limitation on hardware, software and network 			
Requirements for embedded computers	I	Objective	Understanding the basic configuration and features of embedded computers. Explain that connectivity, reliability, efficiency and interchangeability are among the requirements for computer systems.	9-1-3-I	
		Prerequisite	9-1-1-I		
		<ul style="list-style-type: none"> - Basic configuration and features - Connectivity, reliability, efficiency and interchangeability 			
Hardware and architectures for embedded computers	I	Objective	Understanding characteristics of typical hardware and architectures for embedded software	9-1-4-I	
		Prerequisite	9-1-3-I		
		<ul style="list-style-type: none"> - Typical CPUs <ul style="list-style-type: none"> • Such as ARM, MIPS, Xscale, PowerPC, SH, M32R, etc. - Typical OS <ul style="list-style-type: none"> • Such as uCLinux, T-Kernel, eCOS, RTLinux, etc. - Typical network connectivity <ul style="list-style-type: none"> • LAN, WiFi, Infrared, low bandwidth wireless, IEEE1394, etc. - Issues and countermeasures specific to embedded systems <ul style="list-style-type: none"> • Power consumption • Small memory - Basic hardware configuration of embedded systems <ul style="list-style-type: none"> • The basic configuration, roles, and features of embedded computer hardware. 			
		Objective	Understanding embedded hardware architectures including CPU, and flash memory system.		9-1-4-II
		Prerequisite			

		<ul style="list-style-type: none"> - Architecture of CPUs <ul style="list-style-type: none"> • Internal CPU architecture • System buses • Bus data communication • Operations in the CPU. • Memory connections • Instruction Set Architecture - Memory management methods <ul style="list-style-type: none"> • Flash memory management 		
Embedded software basics	I	Objective	Understanding roles and features of embedded software. Such as task and scheduling as a basic part of embedded software processing.	9-1-5-I
		Prerequisite		
		<ul style="list-style-type: none"> - Functions of embedded OS kernels <ul style="list-style-type: none"> • Task and scheduling • Preemption - Resource management <ul style="list-style-type: none"> • Physical memory addressing • Virtual memory addressing 		
Details of embedded OS kernels	II	Objective	Understanding functions, characteristics and roles of embedded OS kernels in detail	9-1-6-II
		Prerequisite		
		<ul style="list-style-type: none"> - Functions required of embedded OS kernels <ul style="list-style-type: none"> • Embedding-specific issues related to task control • Synchronization primitives - Implementation of synchronization <ul style="list-style-type: none"> • Test and set • Semaphores 		
Real-time systems	I	Objective	Understanding necessity, characteristics and processing method and design	9-1-7-I

			of real-time systems.	
		Prerequisite		
			<ul style="list-style-type: none"> - Basics of real-time systems - Design patterns for RT system - Time management - Real time programming - Queuing techniques - Hard real-time - Soft real-time - Device driver and RT - Priority and exclusion order 	
	II	Objective	Understanding the mechanism of the context switch used to perform multiple tasks, and the concept of asynchronous processing and task priority control.	9-1-7-II
		Prerequisite	2-4-5-I (process switching), 2-4-6-II (interrupt)	
			<ul style="list-style-type: none"> - Application demand context switch <ul style="list-style-type: none"> • Preemption-based implementation • Interrupt-based context switch - Application controlled asynchronous / synchronous processing <ul style="list-style-type: none"> • Execution modes • Interrupt priority - Task priority control - Time control methods inside Linux kernel 	
I/O resource management	II	Objective	Understanding concepts of system resource allocation, sharing and management in embedded environment including memory and IO resources.	9-1-8-II
		Prerequisite	2-2-*-II (memory mgmt of Linux), 2-4-*-II (semaphore)	

		<ul style="list-style-type: none"> - Management bus such as I2C, SMB, smart battery - I/O connection and buses such as Universal Serial Bus, IEEE1394, GPIB etc. - Special devices such as sensors, camera, motor and actuators - Communication devices such as WCDMA, GSM, Bluetooth - User interface devices such as touch panel, keypad, 3D accelerometer, thermometer etc. 		
Special memory devices management	II	Objectives	Understanding characteristics of memory devices specific to embedded systems.	9-1-9-II
		Prerequisite	2-2-*-II (memory mgmt of Linux)	
		<ul style="list-style-type: none"> - Flash memory - SSD - Memory cards such as SD card, MMC, memory stick, etc. 		
High reliability implementation	II	Objective	Understanding perspective and technique of improving reliability of embedded application.	9-1-10-II
		Prerequisite		
		<ul style="list-style-type: none"> - High reliability implementation <ul style="list-style-type: none"> • Fault avoidance and fault tolerance • Error detection implementations • Error recovery implementations 		

2.15 Embedded Development Environment

SKILL CATEGORY NAME	Embedded SW		SKILL CATEGORY NO.	9
SKILL NAME	Embedded Development Environment		SKILL NO.	2
TOPICS	LEVEL	SUBTOPICS		CODE NO.
Makefile	II	Objective	To have ability to build own projects by fluently writing Makefiles that can build and install static and/or dynamic libraries.	9-2-1-II
		Prerequisite	<using GCC/G++>	
		<ul style="list-style-type: none"> - Able to write Makefiles <ul style="list-style-type: none"> • Primitives of Makefile (target, dependency, rule) • MACRO and .SUFFIXES • Pre-defined macro • \$<, \$@, \$? • Make depend - Able to write Recursive Makefiles <ul style="list-style-type: none"> • Multiple and hierarchical directories - Able to write Makefiles for building and installing libraries <ul style="list-style-type: none"> • Static libraries • Shared libraries - -fPIC & -shared options in GCC • PATH and Environment variables • Linux commands – ar, ldconfig 		
Managing project	I	Objective	To have the ability to use tools for managing OSS projects collaboratively.	9-2-2-I
		Prerequisite		
		<ul style="list-style-type: none"> - Understanding types of version control systems <ul style="list-style-type: none"> • Revision control system (RCS) • Concurrent Versioning System (CVS, Subversion) 		

		<ul style="list-style-type: none"> • Distributed Versioning System (GIT, BAZZAR) <ul style="list-style-type: none"> - Ability to Use CVS <ul style="list-style-type: none"> • Rationales for CVS • Setting up CVS server • Initializing project • Using CVS commands (check-in, check-out, etc) • Performing backup and recovery - Understand the characteristics of other tools <ul style="list-style-type: none"> • Subversion • Etc 		
GNU tools	I	Objective	To understand the roles of basic GNU tools and their relationships.	9-2-3-I
		Prerequisite	<using GCC/G++>	
		<ul style="list-style-type: none"> - Able to use basic GNU tools for compiler, load, library management - Understand relationships of GNU tools <ul style="list-style-type: none"> • Roles and relationships in gcc, make, automake, autoconf, //GDB 		
	II	Objective	To have ability to use advanced GNU tools for releasing and managing source codes.	9-2-3-II
Prerequisite		9-2-1-II		
<ul style="list-style-type: none"> - Able to use autoconf <ul style="list-style-type: none"> • Rationales for autoconf • Configure script • Configure.ac • Macros - Able to use automake <ul style="list-style-type: none"> • Rationales for automake • Makefile.in • Makefile.am • Recursive Makefile.am 				

		- Use more tools – gcov, ctags, cscope	
Building cross development environment	I	Objective	To have ability to build cross development environments.
		Prerequisite	
		<ul style="list-style-type: none"> - Understand frameworks of cross development environment <ul style="list-style-type: none"> • Target and Host • Connections between target & host - Able to use minicom <ul style="list-style-type: none"> • Connecting serial connection • Setting up minicom – baud rate, parity, etc. - Able to use NFS <ul style="list-style-type: none"> • Understanding NFS • Setting NFS between host and target - Able to use binutils and cross compiler <ul style="list-style-type: none"> • Target compilers - Gas, gcc • Lib and includes for target machine • Setting environment variables (INC, LD_LIB, CC, PATH, etc) for cross compiler 	
Debugging with GDB	I	Objective	To have ability to debug source code using debugging tools such as GDB.
		Prerequisite	<using GCC/G++>
		<ul style="list-style-type: none"> - Able to use debugging tool (GDB) <ul style="list-style-type: none"> • Using Gcc options • Using GDB commands – Run, Step, Next, Trace, Break, Watch point, Inspection for variable and register • Debugging multithread programs • Debugging libraries 	
Bug tracking	I	Objective	To have ability to install and use bug tracking tools.
		Prerequisite	
		<ul style="list-style-type: none"> - Able to use basic operations of Bugzilla 	

		<ul style="list-style-type: none"> - Able to install Bugzilla server - Understand other bug tracking tools <ul style="list-style-type: none"> • Issue tracking – Mantis, Trac 	
Remote debugging	I	Objective	To have ability to setup remote debugging environment.
		Prerequisite	
		<ul style="list-style-type: none"> - Able to setup remote debugging environment with GDB <ul style="list-style-type: none"> • Concept of remote GDB protocol • Running GDB server on target machine • Attaching • Command-line arguments for GDB server - Able to use remote debugging environment with GDB <ul style="list-style-type: none"> • Connecting to GDB server • Sending files to remote target • Monitor command to GDB server • Remote configuration 	9-2-7-I

2.16 Device Driver Development

SKILL CATEGORY NAME	Embedded Software		SKILL CATEGORY NO.	9
SKILL NAME	Device Driver Development		SKILL NO.	5
TOPICS	LEVEL	SUBTOPICS		CODE NO.
Linux kernel	I	Objective	Understand the basics of Linux Architecture.	CODE SHARE
	II	Objective	Use kernel primitive functions.	9-5-1-II
		Prerequisite		
		<ul style="list-style-type: none"> - Use various kernel memory primitives including kmalloc() / kfree(), vmalloc() / vfree(), alloc_page() / free_page(), slab primitives - Understand the process control primitives such as schedule(), sleep() and wakeup() - Understand kobject - Use primitive functions for /proc, /sys file system - Use the macros and primitives to handle kernel lists and queues - Understand kernel timer data structures and primitives 		
Kernel debugging	I	Objective	Setup remote debugging environment and debug.	CODE SHARE
	II	Objective	Monitor and debug kernel remotely.	9-5-2-II
		Prerequisite		

			<ul style="list-style-type: none"> - Monitor kernel event and behavior with /proc, /sys, printk() - Use KGDB to trace kernel code, watch kernel data - Debug LKM (loadable kernel module) - Understand and use the capability of ICE (in circuit emulator) to debug kernel 	
Kernel configuration	I	Objective	Configure Linux Kernel.	9-5-3-I
		Prerequisite		
			<ul style="list-style-type: none"> - Configure Linux Kernel with kconfig system - Build kernel image (vmlinux, bzImage) both in source directory and separate object directory - Configure, build and install kernel module - Configure Linux kernel for remote debugging with KGDB (serial or Ethernet) - Configure Linux kernel for 	
	II	Objective	Understand the method to add new items to kernel.	9-5-3-II
	Prerequisite			
		<ul style="list-style-type: none"> - Understand the method to add new items to kernel - Modify kconfig system - Configure, build and install external kernel module 		
Character device driver	II	Objective	Implement character device driver as a kernel module.	9-5-4-II
		Prerequisite		

		<ul style="list-style-type: none"> - Register and unregister character device driver with specified major, minor number - Implement system call API such as open(), read(), write(), ioctl(), release() - Implement data availability API such as poll(), fasync() - Implement a device driver such as a Linux kernel module 		
	III	Objective	Implement device driver which uses interrupt and/or DMA.	9-5-4-III
		Prerequisite		
		<ul style="list-style-type: none"> - Implement device drive which uses hardware interrupt by setting up IRQ - Understand execution context (user / kernel / interrupt) - Understand the do's and don'ts in ISR - Understand and use delayed execution such as Top Half/Bottom Half 		
Other device drivers	III	Objective	Implement all kinds of device drivers.	9-5-5-III
		Prerequisite		
		<ul style="list-style-type: none"> - Implement the bus drivers including I2C, SPI, PCI, PCMCIA/CF - Understand USB software architecture and implement USB class driver - Implement block device driver - Implement network interface driver - Implement video frame buffer driver - Implement audio driver using open sound system - Implement camera driver for video for Linux - Implement device drivers as a user space application 		

Building target system	I	Objective	Configure boot loaders and manage boot sequences, Build root file system.	9-5-6-I
		Prerequisite	Linux system management (2-3-?-I)	
		<ul style="list-style-type: none"> - Configure bootloader such as uBoot, Redboot, and etc. - Use flash commands to fuse bootloader/kernel/root file system - Configure DHCP/BOOTP/NFS for network boot - Use boot load command or specify kernel options for network boot - Build root file system from scratch - Build Iniramfs file system - Configure, build and install busybox - Configure mtd device and use it as Linux file systems 		
Kernel synchronization primitives	II	Objective	Implement device driver with concurrent access.	9-5-7-II
		Prerequisite		
		<ul style="list-style-type: none"> - Implement process level synchronization using sleep() / wakeup() / wait_event() - Implement mutual exclusion with kernel mutex primitives such as spin lock, read write lock, atomic operations - Understand when spin_lock_irq() is required - Use up/down, completion semaphore 		
Kernel thread	I			(code share) 2-2-6-I
	II	Objective	Implement kernel thread.	9-5-8-II
Prerequisite				

		<ul style="list-style-type: none"> - Create and daemonize a kernel thread - Understand and use softirq/tasklet 		
Power management	III	Objective	Implement power efficient device driver.	9-5-9-III
		Prerequisite		
		<ul style="list-style-type: none"> - Understand Linux power management architecture <ul style="list-style-type: none"> • ACPI states - Understand standby and wakeup mechanism - Use ACPI to manage power consumption 		

Blank Page(Don't remove)

Appendix A. Sample Courses

A-1. Sample Course for OSS based Embedded SW Developer

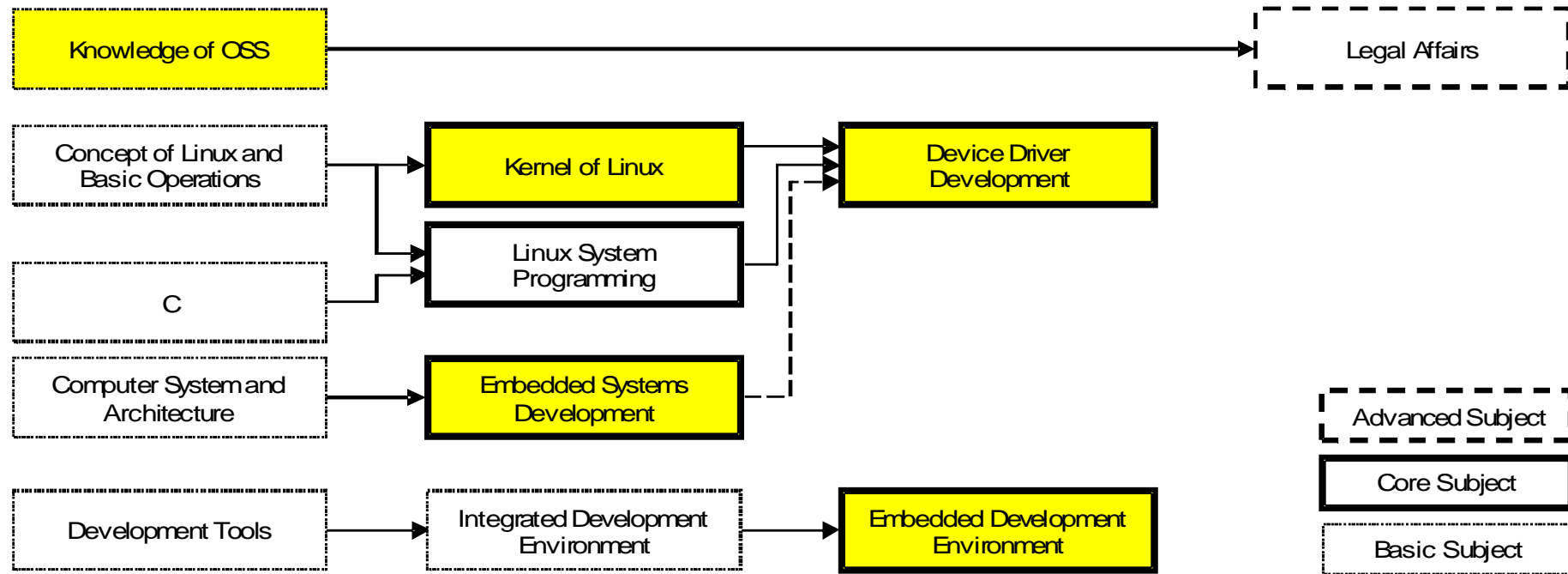


Figure 6. Block diagram of the sample course for OSS-based embedded SW developer

Figure 6 shows the block diagrams which contain 12 subjects: 5 basic classes, 5 core classes and 1 advanced class. Each class is composed of units, i.e., the set of leveled subtopics in a same topic. For example, the core classes are a set of units, each of which is considered crucial for this sample course. Figure 7 shows the selections of the units of each class. For example, the class “Kernel of Linux” consists of units such as Introduction (2-2-1-I), Scheduling (2-2-2-I), Interrupt (2-2-4-I), System Calls (2-2-5-I), Process Management (2-2-6-*), Memory Management (2-2-7-1), and File Systems (2-2-8-I).

Knowledge of OSS	Kernel of Linux	Embedded System Development	Device Driver Development	Embedded Development Environment
Introduction (1-1-1-*) History (1-1-2-I) Use of OSS (1-1-8-I) Community (1-1-10-I) OSS sites (1-1-11-II) OSS OS deployment (1-1-12-II)	Introduction (2-2-1-I) Scheduling (2-2-2-I) Interrupt (2-2-4-I) System Calls (2-2-5-I) Process mgmt (2-2-6-*) Memory mgmt (2-2-7-1) File systems (2-2-8-I)	Task and Scheduling (9-1-1-II) Resource mgmt (9-1-2-II)	Linux Kernel (9-5-1-II) Kernel Debug (9-5-2-II) Kernel Config (9-5-3-*) Character Dev. (9-5-4-II) Building Target (9-5-7-I) Kernel Synch (9-5-9-II) Kernel Thread (9-5-10-II)	Project Building (9-2-1-II) Mgmt Project (9-2-2-I) Cross Dev. Env (9-2-4-I) Remote Debugging (9-2-7-I)

Figure 7. Selection of units for each class

A-2 Sample course for Web based Enterprise Application Developer by OSS

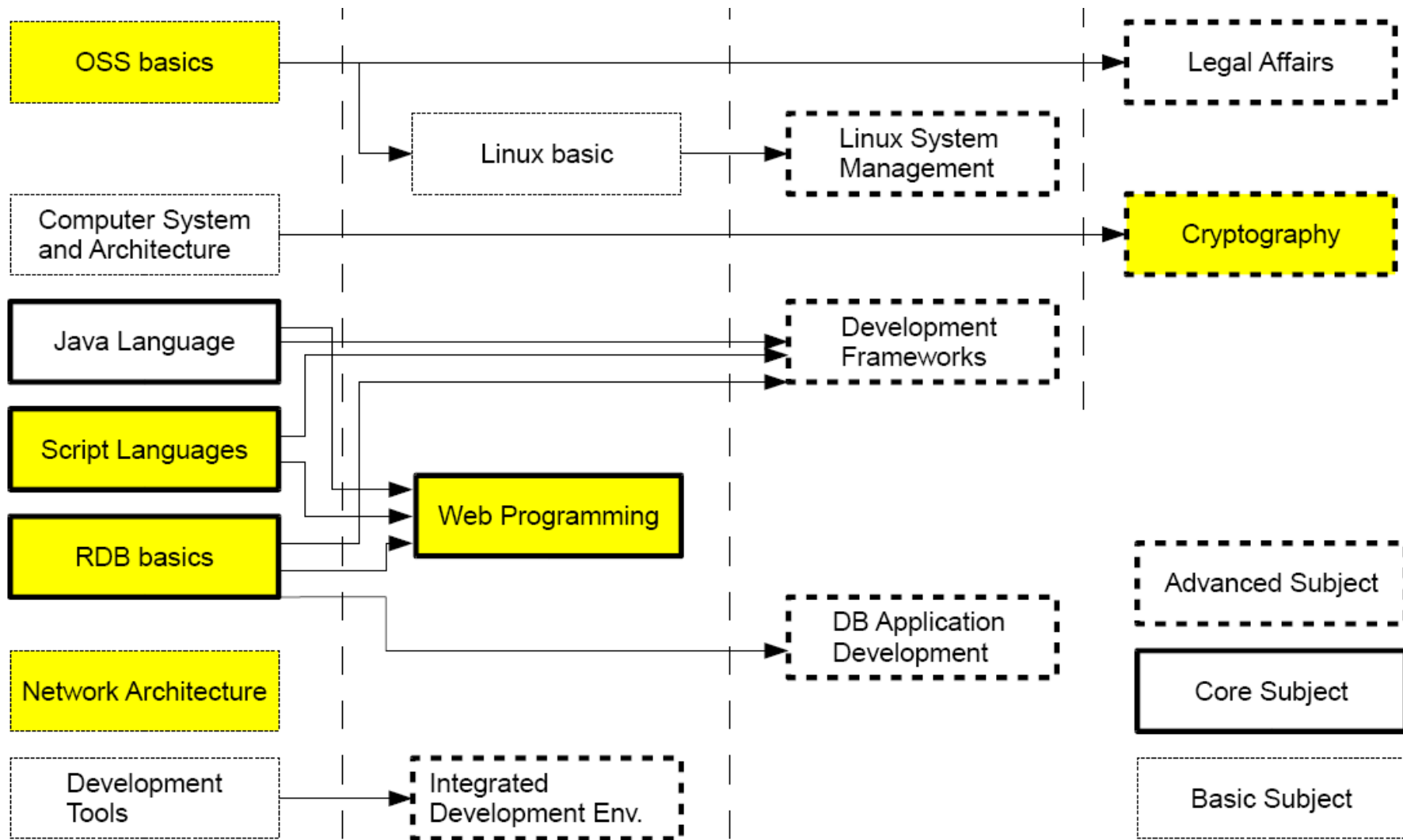


Figure 8. Block diagram of the sample course for Web based Enterprise Application Developer by OSS.

OSS basics	Cryptography	Script Languages	Web Programming	RDB basics	Network Architecture
<ul style="list-style-type: none"> •Introduction to OSS 1-1-1-* •History of UNIX and Linux 1-1-2-* •OSS servers 1-1-3-* •OSS development tools 1-1-4-* •OSS standardization 1-1-6-* •OSS server applications 1-1-7-* •Use of OSS 1-1-8-* •Web system development 1-1-9-* •OSS communities 1-1-10-* •(OSS business 1-1-10-*) •(Earning OSS information 1-1-11-*) •(Deployment of OSS operating system 1-1-12-*) •Deploying server application 1-1-13-* •Deployment of OSS server 1-1-14-* •(Deployment of OSS server application 1-1-15-*) 	<ul style="list-style-type: none"> •Security functions and cryptography positioning 7-1-1-* •Cryptography systems/common key cipher systems 7-1-2-* •Cryptography systems/public key cipher systems 7-1-3-* •Cryptography Toolkits 7-1-4-* •Digital signature 7-1-5-* •OSS utilization scenes 7-1-6-* •Authentication 7-1-8-* •SSH 7-1-10-* •SSL/TLS 7-1-11-* •PGP 7-1-13-* 	<ul style="list-style-type: none"> •History of Script Languages 4-4-1-* •Understanding Web architecture and web 2.0 technology 4-4-2-* •Fundamental structure of Perl 4-4-3-* •Fundamental structure of PHP 4-4-4-* •Fundamental structure of Python 4-4-5-* •Fundamental structure of Ruby 4-4-6-* •Ruby Object-oriented programming 4-4-7-* •Ruby Embedded classes 4-4-8-* •Concept of Ruby on Rails 4-4-9-* •Database application development 4-4-10-* •Outline of DOM 4-4-13-* •Fundamental structure of Javascript 4-4-14-* 	<ul style="list-style-type: none"> •Understanding Web architecture and web 2.0 technology 4-6-1-* •Markup Language for Web Programming 4-6-2-* •Web programming with PHP 4-6-3-* •CGI programming with Perl 4-6-4-* •Internet Programming with Python 4-6-5-* •Web programming with Ruby on Rails 4-6-6-* •Java Programming 4-6-7-* •Web programming with JSP/Servlet 4-6-8-* •XML programming with Java 4-6-9-* •Developing Java Web Services 4-6-10-* •Web Application Development with Ajax 4-6-11-* •Web 2.0 programming with Java 4-6-12-* •Internet Programming with Google APP Engine 4-6-13-* 	<ul style="list-style-type: none"> •Basic database theory 8-1-1-* •Fundamental knowledge of RDBMSs 8-1-2-* •Fundamental concepts of transactions 8-1-3-* •Database components 8-1-4-* •Database access in SQL 8-1-11-* 	<ul style="list-style-type: none"> •The concept and mechanism of open networks 3-1-1-* •The mechanism of TCP applications 3-1-10-*

Figure 9. Selection of units for each class

Appendix B (informative)

This appendix provides the skill-sets not covered in Chapter 2. The purpose of providing this appendix is to show our ongoing efforts toward the full coverage of 38 skill-set definitions. Thus, please treat this informative part with caution as it may be altered without notification.

Table 1. Category of OSS knowledge and skill-sets (repeat)

Category	WG2 OSS Skills	Code
Basic	Knowledge of OSS	1-1
	Legal Affairs	1-2
	Computer System and Architecture	1-3
	Distributed Architecture	1-4
System	Concept of Linux and Basic Operations	2-1
	Kernel of Linux	2-2
	Linux System Management	2-3
	Linux System Programming	2-4
	Network Server Management	2-5
	Cluster System Architecture	2-6
	Concurrent System Programming	2-7
	Java EE Application Server	2-8
Network	Network Architecture	3-1
	Network Management	3-2
	Network Programming	3-3
Programming	Java	4-1
	C	4-2
	C++	4-3
	Script Language	4-4
	GUI	4-5
	Web Programming	4-6
Multimedia System	Multimedia Service Platform	5-1
	Multimedia Programming	5-2
Development System	Development Frameworks	6-1
	Development Tools	6-2
	Integrated Development Environment	6-3
	Software Testing	6-4
Security	Fundamentals of Cryptography	7-1
	Network Security	7-2
	OS Security	7-3
RDB	Basic Skills in RDB	8-1
	RDB System Management	8-2
	RDB Applications Development	8-3
Embedded Software	Embedded System Development	9-1
	Embedded Development Environment	9-2
	Embedded Application Development	9-3
	Embedded System Optimization	9-4
	Device Driver Development	9-5

* Shaded Skills are covered in Section 2.

SKILL CATEGORY NAME	Basic		SKILL CATEGORY NO.	1
SKILL NAME	Legal Affairs		SKILL NO.	2
TOPICS	LEVEL	SUBTOPICS	CODE NO.	
Meaning of OSS licenses	I	<ul style="list-style-type: none"> - Clarify the legal position of OSS licenses (license agreements). - Explain how effective licenses are and what licenses mean to such parties as providers, users, and distributors. 	1-2-1-I	
Selection of OSS license	I	<ul style="list-style-type: none"> - Show the features of typical OSS licenses (GPL, BSD, MPL). - Explain the difference between open source software licenses and commercial licenses. - Different licenses in various situations and purposes. 	1-2-2-I	
	II	<ul style="list-style-type: none"> - Usage of multiple licenses on same product. 	1-2-2-II	
Explanation of typical open source licenses [GPL type]	I	<ul style="list-style-type: none"> - Copyright and copyleft. - Clarify the relation between software copyright and license. Also explain the background to the appearance of the copyleft. Address the relation between the basic concept of the copyleft and GPL, which embodies the concept. - Feature of GPL and the significance of GPL v3 rev. - Describe the definition and history of GNU General Public License (GPL). Also explain the points and significance of GPL v3, which was revised in 2007. - Various problems concerning GPL. - Show that GPL has spread so forcibly that GPL is said to be causing “GPL contamination” and that it is difficult for GPL to coexist with software otherwise licensed. Present actual cases of GPL infringement. 	1-2-3-I	
Explanation of typical open source licenses [MPL type]	I	<ul style="list-style-type: none"> - Features of MPL, and MPL and software patents. - Clarify the features of Mozilla Public License (MPL). Explain MPL employs the bazaar system, which characterizes OSS, and that considerations were given to its coexistence with commercial software. Also point out that MPL is more closely related to patents. 	1-2-4-I	

Explanation of typical open source licenses [BSD type]	I	<ul style="list-style-type: none"> - Advantages and disadvantages of BSDL and Apache licenses. - Explain the features of BSD License and Apache Software License, and their pros-and-cons resulting from their least restriction. Introduce the amended BSD License, referring to well-known “advertisement provision” under the BSD License. 	1-2-5-I
The outline of intellectual property	I	<ul style="list-style-type: none"> - Copyright of software. - The basic idea of copyright as an intellectual property right. - The fundamental idea of the relations of copyrights with the software industry. - The definition and substance of copyright. 	1-2-6-I
	II	<ul style="list-style-type: none"> - Problems and trends concerning software patents. - The basic idea of patent as an intellectual property right. - The definition of patent, patent systems, and their present status in various countries. - Software patents and innovation. 	1-2-6-II
	III	<ul style="list-style-type: none"> - Other intellectual property related to software. - Trademark. - Model utility right. - Design right. - Trends in the software industry besides copyrights and patents. - Introduce topics about these rights, including design right infringement. 	1-2-6-III
Points to be considered from the perspective of intellectual property at the time of using OSS	II		
Points to be considered from a legal perspective other than intellectual property at the time of using OSS	II		
Legal risk management by	II		

corporations/organizations etc.			
Businesses prepared for legal risks	III		
Legal risk reduction measures to be considered by OSS development communities	III		
Legal risk reduction measures to be considered by OSS- business related corporations	III		
Examples of lawsuits/problems in connection with OSS intellectual property issues	III		
Controversy over patents for software	III		
Guidelines for patent applications associated with intellectual property	III		

SKILL CATEGORY NAME	Basic		SKILL CATEGORY NO.	1
SKILL NAME	Computer System and Architecture		SKILL NO.	3
TOPICS	LEVEL	SUBTOPICS	CODE NO.	
The basics of computer architecture	I	<ul style="list-style-type: none"> - System architecture. - Present the basic idea of the substance of a system and the concepts of a computer system. Show that there are various types of system architectures. Explain the substance of system architecture and the outline of system architecture design. 		
The basics of computer hardware	I	<ul style="list-style-type: none"> - Components of a computer system. - The basic configuration of the computer system. - The role and features of each component. - The history of the development of the computer system. - Relation to OSS. - OSS compatibility with device drivers. - Open BIOS. - The disclosure of technical information of hardware. 		
	II	<ul style="list-style-type: none"> - Evaluation indicators for the computer system. - The performance of computer hardware. - Connectivity (plug-and-play, device driver). - Reliability (failure rate, MTTR, MTBF). - Efficiency (performance and throughput, performance evaluation). - Compatibility (standardization of interfaces, etc.). 		
The basics of CPU/memory architecture	I	<ul style="list-style-type: none"> - Basic architecture and role of CPU. - The basic architecture, features of CPU and its history. - Introduce RISC and CISC as typical architectures. - Some types of pipeline processing and multiprocessor architectures as a 		

		<p>mechanism to achieve faster operation.</p> <ul style="list-style-type: none"> - Memory system. - Memory configuration and storage hierarchy. - The basic architecture and role of memory chips and their history. - Introduce the role of a memory chip used as a main memory unit, the hierarchic storage system composed of primary and secondary caches, and methods of connecting memory devices with the CPU. - The role of each memory unit in the computer. 	
Basic architecture and roles of peripheral devices	I	<ul style="list-style-type: none"> - Various types of peripheral devices. - Explain their basic architectures, roles, features, and histories. - FD, MO, and CD-R as auxiliary storage devices. - Keyboard, mouse, scanner, tablet as input devices. - Display and printer as output devices. 	
The basics of interface technology	I	<ul style="list-style-type: none"> - Features of various interfaces. - The basic architecture, role, and features of interfaces and their history. - USB, IEEE1394, and eSATA as hot-pluggable interfaces. - SCSI, IDE, PCI, and serial types as legacy interfaces. 	
	II	<ul style="list-style-type: none"> - Fiber channel, iSCSI. 	
Software architecture		<ul style="list-style-type: none"> - Components of computer software. - The operating system (OS), middleware, and applications, explain their function, architecture, and history. - Implementation of OSS version of software. 	
OS architecture		<ul style="list-style-type: none"> - The basic architecture, role, features, and history of OS. - Outline of the current status of implementation of OSS and its mechanism. - Resource management of files, networks, input-output interfaces, and other resources. 	
Types and features of		<ul style="list-style-type: none"> - The basic architecture, role, and features of middleware. 	

middleware		<ul style="list-style-type: none"> - Web servers. - Application servers. - Databases. - The current status of implementation of OSS middleware. - Introduce typical OSS compatibles with such middleware. 	
Computer system configuration			
Examples of system architecture utilization			
Web system architecture			
Infrastructure design casework using OSS			
Open source system architecture creation			
Hardware as an OSS operating environment			
Future trends in open source architecture			

SKILL CATEGORY NAME	Basic		SKILL CATEGORY NO.	1
SKILL NAME	Distributed Architecture		SKILL NO.	4
TOPICS	LEVEL	SUBTOPICS		CODE NO.
Distributed architecture	II			
Name management	II			
Duplicate management	II			
Fault tolerance	II			
Secure channels	II			
CORBA	II			
Web services	II			
Distributed transactions	II			
Peer-to-peer architecture	II			
Mobility	II			
Consistency	II			
Large-scale distributed systems	II			

SKILL CATEGORY NAME	System		SKILL CATEGORY NO.	2
SKILL NAME	Concept of Linux and Basic Operations		SKILL NO.	1
TOPICS	LEVEL	SUBTOPICS	CODE NO.	
The outline of Linux	I	<ul style="list-style-type: none"> - UNIX and Linux, and the history of Linux development. - The role of operating system (OS) and the basic idea of UNIX. - The relations between UNIX and Linux. - The history of Linux development. 		
	II	<ul style="list-style-type: none"> - Linux development model and points to remember in using the model. - The development style using the bazaar model—a feature of the Linux development model. Concerning the use of Linux and applications running on Linux, present cases related to licenses and business. Explain the general use of Linux applications. - The substance, definitions, and features of Linux distribution. - Examples of typical distribution systems. - The features and types of each distribution system. 		
System management	I	<ul style="list-style-type: none"> - The concept of the user. - The log-in and log-out procedure. - The basic operation of UNIX/Linux using the B shell (bash). - The basic idea of dialogs with OS using a shell. - Online manuals. - Man command. - Info command. - Practical operations such as top, ps etc. - Editor (vi emacs). - Pager (more less). - Input-output stream. 		

		<ul style="list-style-type: none"> - The idea of redirect and pipe. - Tree structure of the file system and its operation. - The file system as a tree structure. - Many types of file systems. - The concept and features of the file system. - Typical commands and their usage in basic file/ directory operation. 	
Multitasking		<ul style="list-style-type: none"> - The idea of multitasking as a basis for processing on UNIX/Linux. - The difference between single-tasking and multitasking. - The difference between single-users and multi-users. - Time-sharing systems and multitasking. 	
File systems	I	<ul style="list-style-type: none"> - Multiple users and groups. - Permission by the users and groups. 	
	II	<ul style="list-style-type: none"> - Access lists (POSIX ACL). 	
Data saving and backup			
Shell scripts and development environment			
The basics of networks			

SKILL CATEGORY NAME	System		SKILL CATEGORY NO.	2
SKILL NAME	Linux System Management		SKILL NO.	3
TOPICS	LEVEL	SUBTOPICS	CODE NO.	
The outline of Linux system management work	I	<ul style="list-style-type: none"> - Role and management operations of the system administrator. - Outline of systems management operations. - The purpose, necessity, and types of systems management operations. - The roles and authority of the system administrator. - The mindset required of the administrator. 		
Linux system and server management	I	<ul style="list-style-type: none"> - Implementation of a Linux system. - Linux implementation tasks. - An actual implementation procedure. - The setup procedure. - The implementation and management procedure for application packages. - Precautions to be taken in implementation. - Startup and shutdown of the Linux system. - The startup and shutdown procedure for the Linux system. - Init process. - Runlevel. - Basic knowledge of the startup and shutdown of services. 		
Linux system and file/disk management				
Linux system and user management				
Linux system management, and backup and log operation management				
Linux system and resource				

management			
Linux system and kernel management			
Linux system and network management			
Linux system and routing management			
Linux system management, and DHCP construction and operation			
Linux system management, and FTP construction and operation			
Linux system management, and NFS construction and operation			
Linux system management, and Samba building and operation			
Linux system management and basic operation work troubleshooting			
Linux system management and network troubleshooting			

SKILL CATEGORY NAME	System		SKILL CATEGORY NO.	2
SKILL NAME	Network Server Management		SKILL NO.	5
TOPICS	LEVEL	SUBTOPICS	CODE NO.	
Functions and features of network servers	I	<ul style="list-style-type: none"> - Role of the server and communication protocols. - The role and basic mechanism of the servers. - The functions and features of network servers. - Communication protocols. - Basic concepts of client-server systems. - Meaning of the network address, domain name, and host. - The structure of the network address (IP address). - Domain name and host name. - Network byte order. - Uniform Resource Identifier (URI). 		
Server system installation	I	<ul style="list-style-type: none"> - An orderly outline and work procedure for the introduction of general network servers. - The acquisition of OS and required software packages, implementation, configuration of the startup environment, network configurations, service content settings, and the startup and shutdown of the server system. 		
Name server installation	I	<ul style="list-style-type: none"> - Give an outline of Domain Name System (DNS) and the mechanism of the DNS server and the DNS protocol to offer DNS services. Provide related information concerning such topics as the history of, and background to, DNS server implementation based on OSS. - The procedure for introducing, configuring, and setting up a Linux-based name server. - The configuration file. 		
Web server installation	I	<ul style="list-style-type: none"> - Mechanism of the Web server. - The functions and roles of the Web server. 		

		<ul style="list-style-type: none"> - The execution and extension of applications with CGI. - The history of, and background to, Web server implementation with OSS. - The outline of Hyper-Text Transport Protocol (HTTP) and communication methods. - Procedure for introducing, configuring, and setting up a Linux-based Web server. - The configuration file. 	
Super server installation	II	<ul style="list-style-type: none"> - inetd - xinetd - tcpwrapper 	
Proxy server installation	II	<ul style="list-style-type: none"> - Squid - Apache proxy 	
	III	<ul style="list-style-type: none"> - Reverse proxy. - Proxy authentication. - NTLM authentication. 	
Details of other network server installation work and its work procedures	II		
Routing processing and filtering processing implementation by network servers	II		
Internet connection via network servers	II		
Server operation management tasks	II		
Log management details and procedures	II		

Linux server security	II		
Linux service security	II		
Functions and implementation of secure OSes	II		

SKILL CATEGORY NAME	System		SKILL CATEGORY NO.	2
SKILL NAME	Cluster System Architecture		SKILL NO.	6
TOPICS	LEVEL	SUBTOPICS	CODE NO.	
A general introduction to cluster systems	I	<ul style="list-style-type: none"> - The outline and features of the High Availability (HA) cluster and the High Performance Computing (HPC) cluster. - Compare the two cluster systems in terms of the purpose and position. - The load distribution technology as a key HA cluster technology, explain its concepts, types, configuration, and components. Introduce the practical application of the load balancer, which enables load distribution. Describe the configuration and features of each type of load balancer. - Introduce typical algorithms, such as a round robin algorithm, that enable load distribution. Explain a practical procedure for the algorithm. Address the technique to monitor servers for their state of life or death. Explain tasks to be performed during the course of operation of the HA cluster. 		
HA clusters	I	<ul style="list-style-type: none"> - Realization of a load balancer by the Linux Virtual Server (LVS). - The components of the system. Provide a practical procedure for building an IPVS kernel, ipvsadm and keepalived, a testing environment, and a Direct Server Return (DSR) system. - Redundancy by the Virtual Router Redundancy Protocol (VRRP). - The Virtual Router Redundancy Protocol (VRRP), which is used to provide redundancy for routers. Explain the ways to configure a network and make keepalived settings achieve redundancy. Also give practical advice on keepalived management. 		
Computer simulation		<ul style="list-style-type: none"> - Supercomputer: history and realization by clusters. - The outline, history, and recent trend of the supercomputer as part of the scientific and technological infrastructure. Explain how an increasing number of supercomputers have been developed in recent years by the use of clusters. Introduce some PC-cluster projects. 		

A general introduction to parallel programming	I	<ul style="list-style-type: none"> - Types and configurations of parallel computers. - The principles of parallel programming. - Types of parallel computers and differences between them in configuration for processor, memory binding method, and bus topology. - Parallel processing and the basics of programming for parallelization. - Typical algorithms to demonstrate parallelization techniques. - The parallel programming environment and tools available for parallel programming. 	
Beowulf PC cluster construction	II		
SCore clusters	II		
PC cluster-related technology	II		
Grid computing	II		

SKILL CATEGORY NAME	System		SKILL CATEGORY NO.	2
SKILL NAME	Concurrent System Programming		SKILL NO.	7
TOPICS	LEVEL	SUBTOPICS	CODE NO.	
Introduction to concurrent systems (incl. HA, HPC cluster)	I	<ul style="list-style-type: none"> - Introduction to concurrent system: <ul style="list-style-type: none"> • SMP system • HA, HPC cluster system • Grid computing 		
Parallel processes	II	<ul style="list-style-type: none"> - Parallel process. - Concurrency. - Synchronization. - Mutual Exclusion. - Deadlock. - Generic synchronization primitives: <ul style="list-style-type: none"> • Interrupt control • Mutex primitives (lock, semaphore) • Monitor 		
Linux IPC mechanisms	II	<ul style="list-style-type: none"> - Using Linux IPC mechanisms: <ul style="list-style-type: none"> • Message queue • Shared memory • Pipe - UNIX socket 		
Lightweight processes – POSIX threads	II	<ul style="list-style-type: none"> - pthread programming: <ul style="list-style-type: none"> • Create / exit • Join - Shared data handling with pthread: <ul style="list-style-type: none"> • Mutex • Semaphore 		

		<ul style="list-style-type: none"> • Conditional wait 	
Parallel programming	III	<ul style="list-style-type: none"> - Parallel programming with standard interface: <ul style="list-style-type: none"> • openMP • MPI - Monitoring and debugging of parallel programs. 	
Asynchronous events	II	<ul style="list-style-type: none"> - Signal handling. - setjmp/longjmp 	

SKILL CATEGORY NAME	System		SKILL CATEGORY NO.	2
SKILL NAME	Java EE Application Server		SKILL NO.	8
TOPICS	LEVEL	SUBTOPICS		CODE NO
Understanding Java EE	I	Objective	Understanding the basic architecture and functions of the JavaEE and acquiring the skills for installing and configuring JavaEE application servers.	2-8-1-I
		Prerequisite		
		<ul style="list-style-type: none"> - Java EE Architecture overview - Guide to OSS JavaEE Application Server: <ul style="list-style-type: none"> • Apache HTTP Server, Tomcat, JBoss, Gernimo, JOnAS, and so on - Installing and configuring Apache, Tomcat, JBoss 		
Web programming with JSP/Servlet	II	Objective	Understanding the structure of programming with JSP/servlet, and acquiring the skills to develop Web applications and to integrate those with database into JSP pages	2-8-2-II
		Prerequisite	2-8-1-I	
		<ul style="list-style-type: none"> - Developing a basic Java Servlet - Developing a View Component - Developing a Controller Component - Developing Dynamic Forms - Sharing Application Resources Using the Servlet Context - Designing the Business Tier - Developing Web Applications Using Struts - Developing Web Applications Using Session Management - Using Filters in Web Applications 		

Web programming with JSP/Servlet	II	<ul style="list-style-type: none"> - Integrating Web Applications With Databases: <ul style="list-style-type: none"> • JDBC overview • JDBC Statement/Prepared Statement • JDBC Transaction • JDBC Exception • Connection Pool • Introduction to JDBC 2.0 and 3.0 - Developing JSP Pages - Developing JSP Pages Using Custom Tags - Developing Web Applications Using Struts Action Forms - Building Reusable Web Presentation Components 		2-8-2-II
EJB programming	II	Objective	Understanding the concept of developing EJB programming and acquiring the skills to develop EJB 3.0 components and other related components such as Model2-style Web components Developing Transactional components, Transactional components, JMS components, etc.	2-8-3-II
		Prerequisite	2-8-1-I	
		<ul style="list-style-type: none"> - Overview of developing Java EE components - Developing EJB 3.0 components - Developing Entity class using Java Persistence API(JPA) - Developing Model2-style Web components - Developing Transactional component - Developing Java Messaging Service (JMS) components - Developing Web services component using EJB 3.0 components - Overview of Security API 		

JSF programming	III	Objective	Understanding features and architecture of JSF, and acquiring the skills to develop JSF applications and to integrate with Apache shale, struts, and JBoss.	2-8-4-III
		Prerequisite	2-8-3-II, 2-8-2-II	
		<ul style="list-style-type: none"> - Introduction to JSF - Standard Features: <ul style="list-style-type: none"> • Creating backing beans and using managed beans • Exploring the standard components • Internationalization, validators, and converters • Using Facelets - Application Development: <ul style="list-style-type: none"> • Developing with JSF technology • Inside the JSF Architecture • Writing custom components, validators, and converters - Extensions and Integration: <ul style="list-style-type: none"> • Component frameworks overview • Using Apache Shale • Migrating from Struts • Using JBoss Seam • Summary and future directions 		

Programming with OSS JavaEE Framework	III	Objective	Understanding object-oriented design pattern and OSS JavaEE framework, and acquiring skills for using Spring and ORM frameworks.	2-8-5-III
		Prerequisite	2-8-3-II, 2-8-2-II	
		<ul style="list-style-type: none"> - Understanding Object-oriented Design Patterns - Understanding Software architecture - Understanding JavaEE architecture layer - OSS JavaEE Framework overview - Classification of OSS JavaEE Framework - Supporting Framework(Maven, Log4j, JUnit) overview - Spring Framework overview - DI, SpringMVC, SpringDAO, AOP and so on - ORM Framework overview - ORM Framework (Hibernate, iBatis) overview 		

J2EE on JBoss	III	Objective	Acquiring skills for developing JBoss applications with emphasis on the deployment, IDE, JNDI, JavaMail, JMX and JMS	2-8-6-III
		Prerequisite	2-8-3-II, 2-8-2-II	
		<ul style="list-style-type: none"> - JBoss: <ul style="list-style-type: none"> • Overview of JBoss Application Server • Highlights of JBoss AS • JBoss AS Architecture • JBoss AS Requirements - Installation - Directory Structure - Starting/Stopping - Deployment: <ul style="list-style-type: none"> • J2EE development-deployment life-cycle • J2EE deployment descriptors • Deployment on JBoss AS • Deployment Dependencies • Hot vs. Cold Deployment • Looking at conf/jboss-service.xml and XMBeans - JBoss-IDE: <ul style="list-style-type: none"> • Overview of JBoss IDE plug-ins for Eclipse • Configuring JBoss IDE • Controlling JBoss AS from Eclipse 		

J2EE on JBoss	III	<ul style="list-style-type: none"> - Web Tier: <ul style="list-style-type: none"> • Java Servlets • Welcome files • Error documents • Introduction to Servlet Filters • Life-cycle • API • Defining and mapping in WEB-INF/Web.xml • Java Server Pages • Tomcat Web Container • Overview of Tomcat • Architecture • Configuration (jboss-service.xml and server.xml) • Tomcat's Web.xml • Serving static content • Virtual hosting • Web access logging - JNDI: <ul style="list-style-type: none"> • Overview of JNDI • Role of JNDI in J2EE • JNDI API • Using JNDI • JNDI resources • JNDI on JBoss AS - JavaMail: <ul style="list-style-type: none"> • Overview of JavaMail • JavaMail API • JavaMail service on JBoss AS • Sending (SMTP) and fetching (IMAP/POP) mail 	2-8-6-III
---------------	-----	--	-----------

J2EE on JBoss	III	<ul style="list-style-type: none"> - JMX: <ul style="list-style-type: none"> • Overview of Java Management Extensions • JMX Architecture (layers) • JMX MBeans • JMX on JBoss AS • JMX Console • Twiddle tool (JMX command-line client) • Developing MBeans through JBoss IDE and XDoclet • Packaging service archive (SAR) files • Web Console • JBoss Monitoring - Class Loading: <ul style="list-style-type: none"> • Java Class Identity • J2EE Class Loading Requirements • Class Loading on JBoss • Default Class Search Order • Scoped Class Search Order • Log4j Issues • Common Problems With Class Loading - Database Integration <ul style="list-style-type: none"> • Managed Database Connection Pools • Referencing database connection pool resources • Installing JDBC drivers • Defining database resources in JBoss AS • Resource Mapping • MySQL example • JBoss embedded Hypersonic database • Detecting DB Connection Leaks 	2-8-6-III
---------------	-----	--	-----------

J2EE on JBoss	III	<ul style="list-style-type: none"> - JMS: <ul style="list-style-type: none"> • Overview of Java Messaging Service • JMS Architecture • JMS Messaging Domains (Point-to-Point, Publish, and Subscribe) • JMS Message Consumption • JMS API • Developing with JMS • JMS on JBoss AS • Configuration • Creating destinations (queue and topic) - EJB - Transactions: <ul style="list-style-type: none"> • Overview of Transactions (ACID properties) • Resource Locking (pessimistic vs. optimistic) • Distributed Transaction Components (JTA API) • Two-phase XA protocol • Heuristic Exceptions • Transactions on JBoss AS • Container-Managed Transactions (CMT) • User Transactions - Web Services <ul style="list-style-type: none"> • Web Services on JBoss AS • Servlet-based Web service example • Creating end-point • Generating descriptors (including WSDL) using WSDP • Packaging and deploying • Creating Web service client 	2-8-6-III
---------------	-----	--	-----------

J2EE on JBoss	III	<ul style="list-style-type: none"> - Security: <ul style="list-style-type: none"> • Filtering Clients by Source (IP address or hostname) • Authentication and Authorization using JAAS • Role-based declarative security • Requiring authentication and authorization in deployment descriptors • JBoss plain-text login module • JBoss database login module (example with MySQL) • Linking to security domain • Securing passwords with MD5 • FORM-based login (including the handling of errors) • Configuring JBoss AS for SSL • Generating SSL certificates • Tomcat's SSL connector • Requiring SSL in Web applications • Securing JBoss AS • Running with low-privileges • File-system security • Securing console applications/tools and services • Running with Java Security Manager • Running behind a firewall 	2-8-6-III
---------------	-----	---	-----------

JBoss Clustering	III	Objective	Acquiring skills for developing JBoss applications with emphasis on JBoss Clustering	2-8-7-III
		Prerequisite	2-8-1-I,2-8-6-III	
		<ul style="list-style-type: none"> - JBoss Clustering <ul style="list-style-type: none"> • Overview of JBoss/JEMS Clustering - JGroups <ul style="list-style-type: none"> • Overview of JGroups • JGroups API • JGroups Protocols • JGroups Troubleshooting - TreeCache <ul style="list-style-type: none"> • TreeCache Architecture • TreeCache API • TreeCache Configuration • TreeCache Loading/Eviction Policies - TreeCache AOP <ul style="list-style-type: none"> • Overview of TreeCache AOP • TreeCache AOP Architecture • TreeCache AOP API • TreeCache AOP Configuration • POJO Instrumentation - EJB Clustering <ul style="list-style-type: none"> • Client Side Load Balancing • Stateful Session Beans • Entity Beans/Hibernate 		

JBoss Clustering	III	<ul style="list-style-type: none"> - JBoss Web Clustering <ul style="list-style-type: none"> • Overview of JBoss Web Clustering • Introduction to Tomcat • HTTP Session Replication • Native Web servers • Clustered SSO - JEMS Clustered Services <ul style="list-style-type: none"> • HA-JNDI • DRM/RPC • Farming • HA-Singleton - HAJMS <ul style="list-style-type: none"> • JBossMQ 	2-8-7-III
------------------	-----	---	-----------

SKILL CATEGORY NAME	Network		SKILL CATEGORY NO.	3
SKILL NAME	Network Management		SKILL NO.	2
TOPICS	LEVEL	SUBTOPICS		CODE NO.
The outline of network system operations				
Individual items and details of network management				
Individual items and details of network capacity management				
Individual items and details of network performance management				
TCP/IP management				
Network server operation management practice				
Network server operation management practice				
Network hardware operation management				
The outline of network Management protocols		<ul style="list-style-type: none"> - snmp - mib 		
Exercising network management using MRTGs				
Network operation design				
Operation management practical procedures and its system				
WAN operation management				
Network fault management				
Network troubleshooting				

SKILL CATEGORY NAME	Network		SKILL CATEGORY NO.	3
SKILL NAME	Network Programming		SKILL NO.	3
TOPICS	LEVEL	SUBTOPICS	CODE NO.	
Data Communication Basic	I	<ul style="list-style-type: none"> - Fundamentals of data communication technologies: <ul style="list-style-type: none"> • Dedicated / switched / virtual circuits • Analog and digital transmission • Bandwidth and latency • Transmission media - Connectivity and inter-networking: <ul style="list-style-type: none"> • Network topologies • LAN equipment • Ethernet / Wireless network - Packet / protocol / interface 	3-3-1-I	
Network architecture and Internet	I	<ul style="list-style-type: none"> - Layered network architecture: <ul style="list-style-type: none"> • OSI 7 layer reference model - Internet architecture: <ul style="list-style-type: none"> • Definition and topologies • Addressing - Domain name system - Internet protocols: <ul style="list-style-type: none"> • TCP/IP protocol stacks • Application level protocols 	3-3-2-I	
Linux network system architecture	I	<ul style="list-style-type: none"> - Linux network system architecture: <ul style="list-style-type: none"> • OSI Layer mapping to Linux network system - Linux network commands: <ul style="list-style-type: none"> • netstat / route / ifconfig / ping 	3-3-3-I	

		<ul style="list-style-type: none"> - Network configuration files under /etc: <ul style="list-style-type: none"> • rc scripts • Super server (xinetd) configuration - Linux network security basics 	
TCP/IP protocol internals	II	<ul style="list-style-type: none"> - TCP/IP protocols: <ul style="list-style-type: none"> • ARP / IP / ICMP / TCP / UDP • TCP flow control / error recovery mechanism - Application layer protocols: <ul style="list-style-type: none"> • telnet / ftp / http / mail 	3-3-4-II
Client server architecture	I	<ul style="list-style-type: none"> - Client - server architecture: <ul style="list-style-type: none"> • Basic operation scenarios • Naming / connection / transfer / disconnection - Stateful / stateless server - Peer-to-peer architecture 	3-3-5-I
TCP Socket programming	II	<ul style="list-style-type: none"> - Using bsd socket interface: <ul style="list-style-type: none"> • Creating and destroy • Byte ordering - TCP client-server programming: <ul style="list-style-type: none"> • connect() • bind(), listen(),accept() • recv() / send() - Name resolving: <ul style="list-style-type: none"> • Hostname and IP address - Service and port number 	3-3-6-II
	III	<ul style="list-style-type: none"> - Socket options 	3-3-6-III
UDP Socket programming	II	<ul style="list-style-type: none"> - UDP message transfer: 	3-3-7-II

		<ul style="list-style-type: none"> • recvfrom() / sendto() - UDP client-server programming: <ul style="list-style-type: none"> • bind() - connect() 	
	III	- UDP multicasting	3-3-7-III
Server programming	II	<ul style="list-style-type: none"> - Supporting multiple servers with fork() - Server using super-server (xinetd) 	3-3-8-II
Synchronous (blocking), asynchronous (non-blocking) interface	II	<ul style="list-style-type: none"> - Blocking I/O, non-blocking I/O - Non-blocking socket programming: <ul style="list-style-type: none"> • select() / pselect() • poll() / ppoll() - Multiple socket management - Multithread socket application 	3-3-9-II
Linux IPC mechanisms	II	<ul style="list-style-type: none"> - Using Linux IPC mechanisms: <ul style="list-style-type: none"> • Message queue • Shared memory • Pipe • UNIX socket 	3-3-10-II
Network programming with security	III	- Programming with OpenSSL	3-3-11-III
Interfacing to OSS P2P, VOIP	III	<ul style="list-style-type: none"> - P2P programming: <ul style="list-style-type: none"> • Programming with gtk-gnutella - VOIP programming - Programming with Openh323 	3-3-12-III

SKILL CATEGORY NAME	Programming		SKILL CATEGORY NO.	4
SKILL NAME	Java		SKILL NO.	1
TOPICS	LEVEL	SUBTOPICS		CODE NO.
The basics of Java				
Fundamental structure of Java language				
Advantages of object-oriented programming				
Procedures for application development in Java				
Network programming in Java				
The outline of Web application development according to Servlet/JSP/JDBC	I			
Database access according to JDBC	II			
MVC models	I			
Application development according to EJB	II			
Object-oriented system analysis/design/implementation on practice technology	II			
Development procedures according to design patterns	II			
Java performance tuning	II			

SKILL CATEGORY NAME	Programming		SKILL CATEGORY NO.	4
SKILL NAME	C++		SKILL NO.	3
TOPICS	LEVEL	SUBTOPICS		CODE NO.
C++ Basic Programming	I	Objective	Understanding basics of object-oriented programming and syntax and functions of C++ language.	4-3-1-I
		Prerequisite	C language basics	
		<ul style="list-style-type: none"> -Class & Object. -Functions and Member functions. -Inheritance & Containment. -Object Pointer, Array & Structure. -Dynamic allocation (new, delete). -Polymorphism (Virtual Function, Overriding). -Exception Handling. -Namespace. -Template. 		
STL	II	Objective	Utilize standard template languages and understanding the object patterns	4-3-2-II
		Prerequisite		
		<ul style="list-style-type: none"> -Container, Sequence, Associative Container & allocator. -Refinement (reflexivity, containment, transitivity). -Trait. -Regular type (assignable, default constructible, equality comparable, less than comparable). -Iterator (trivial iterator, input iterator, output iterator, forward iterator, bidirectional iterator, random access iterator). 		

		-Function Object (generator, adaptable generator, predicate).	
--	--	---	--

SKILL CATEGORY NAME	Programming		SKILL CATEGORY NO.	4
SKILL NAME	GUI		SKILL NO.	5
TOPICS	LEVEL	SUBTOPICS	CODE NO.	
GTK	II	<ul style="list-style-type: none"> - GTK Programming Scheme - Initializing GTK - GtkWidget & GtkWindow & Main Loop - Widget Hierarchy - Widget Properties - Signal & Callbacks, Events - Container Widgets (GtkContainer, Boxes, Panes, Tables, Expanders, Handle Boxes, Notebooks, Event Boxes) - Buttons - Dialogs (Built-in Dialogs, Dialogs with Multiple Pages) - Text View Widget (Scrolled Windows, Text Views, Text Iterators and Marks, Text Tags, Inserting Image & Child Widgets, GtkSourceView) - Tree View Widget (GtkListStore, GtkTreeStore, Referencing Rows, Adding Rows and Handling Selections, Editable Text Renderers, Cell Data Functions, Cell Renderers) - Menus and Toolbars (Pop-up Menus, Keyboard Accelerators, Status Bar Hints, Menu Bars, Toolbars, Toolbar Items, Dynamic Menu Creation, Custom Stock Items). Dynamic User Interfaces (User Interface Design, The Glade User Interface Builder, Using Libglade) - Using gtk_config 	4-5-1-II	
QT	II	<ul style="list-style-type: none"> - Qt programming Scheme (signal, slot, emit, Q_OBJECT) 	4-5-2-II	

		<ul style="list-style-type: none"> - Qt Class Hierarchy - Main Windows (Qmain Window, Menus and Toolbars, Status Bar, Multiple Documents, Splash Screens) - Base Classes (Qobject, Qstring and Other Classes, Inheritance Hierarchy) - Layouts (Manual Layout, Automatic Layout, Splitter, Stacked Layouts) - Dialogs (Modal Dialogs, Non-modal Dialogs, Semi-modal Dialogs, Avoiding Bloated Dialogs, Ready-made Dialogs in Qt) - Qtable Widget & Qtable Widget Item - Item View Classes - Container Classes - Custom Widgets (Qwidget, Qt Designed, Double Buffering) - Events and the Clipboard (Event Loop and Handler, Handling Events, Event Filters, Drag and Drop) - 2D and 3D Graphics - Drag and Drop - QSql Module - Input/Output Interfaces - Threading with Qthread - Handling XML with QtXml - Using qmake - Qt Designer (layout and form design tool) 	
--	--	--	--

SKILL CATEGORY NAME	Multimedia System		SKILL CATEGORY NO.	5
SKILL NAME	Multimedia Service Platform		SKILL NO.	1
TOPICS	LEVEL	SUBTOPICS	CODE NO.	
The basics of Multimedia Service	I	<ul style="list-style-type: none"> - An introduction to Multimedia Service - Multimedia Service Platform Overview - Streaming Server (Broadcasting & VOD/AOD System) - Upload/Download Server - Client software - Multimedia Service Platform Interface 	5-1-1-I	
	II	<ul style="list-style-type: none"> - Codecs: <ul style="list-style-type: none"> • Snow • FFV1 • ATRAC3 • H.261, H.263 and h.264/MPEG-4 AVC • Indeo 2 and 3 • QDesign Music Codec 2, used by many QuickTime movies prior to QuickTime 7 • Smacker video • Sorenson 3 Codec used by many QuickTime movies • Theora (together with Vorbis makes a base for the .ogg format) • Truespeech • TXD • VP5 and VP6 • Vorbis • Windows Media Audio • Some Windows Media Video codecs, including WMV1, WMV2, and WMV3 	5-1-1-II	

		<ul style="list-style-type: none"> - Multimedia File Formats: <ul style="list-style-type: none"> • ASF, AVI, BFI, IFF, RL2, FLV, Matroska, Maxis XA, TXD • MSN Webcam stream, MPEG transport stream 	
Implementation of multimedia service platform using FFmpeg	II	<ul style="list-style-type: none"> - Introduction to FFMPEG - ffmpeg: command line tool to convert one video file format to another - ffserver: HTTP multimedia streaming server for live broadcasts - ffmpeg: simple media player based on SDL and on the FFmpeg libraries - libavcodec: all the FFmpeg audio/video encoders and decoders - libavformat: demuxers and muxers for audio/video container formats - libavutil: helper library containing routines common to different parts of FFmpeg - libpostproc: video post-processing routines - libswscale: video image scaling routines 	5-1-2-II
	III	<ul style="list-style-type: none"> - Multimedia Service Architecture based FFmpeg - Video and Audio grabbing - X11 grabbing - Video and Audio file format conversion - Uploading & Content Management - Broadcasting Service - VOD Service 	5-1-2-III

SKILL CATEGORY NAME	Multimedia System		SKILL CATEGORY NO.	5
SKILL NAME	Multimedia Programming		SKILL NO.	2
TOPICS	LEVEL	SUBTOPICS	CODE NO.	
Understanding Multimedia Programming	I	<ul style="list-style-type: none"> - Multimedia Programming Basic - Multimedia Application Implementation on Web (API Programming Skill) - Display Objects - Text & Image - Geometry Objects (Point, Line, Rectangle, Polygon etc) - Introduce Programming Tools (Video4Linux, Open Sound System, etc) - Introduce RIA (Flash, Flex etc) 	5-2-1-I	
	II	<ul style="list-style-type: none"> - Basics of codecs - Event Processing - Style & Theme - Effects - Data handling & Using XML - Video & Camera handling - Using Audio & Microphone - Optical media (CD, DVD) interface - Print Skill - JPEG library - FFMPEG library 	5-2-1-II	
Video4Linux	II	<ul style="list-style-type: none"> - An introduction to Video4Linux - Graphics cards with TV Tuner and/or Capture facilities 	5-2-2-II	

		<ul style="list-style-type: none"> - TV cards and drivers - Supported tuners - Video via PCI - Video via PCI Express - Video via USB - Firewire - Loopback - Radio - Remote controllers - Scanners - Webcams - Supported hardware vendors 	
	III	<ul style="list-style-type: none"> - An introduction to API - Registration and open() - Basic ioctl() handling - Inputs and Outputs - Colors and formats - Format Negotiation - Basic Frame I/O - Streaming I/O - Controls 	5-2-2-III
Open Sound System	II	<ul style="list-style-type: none"> - An introduction to OSS (Open Sound System) - OSS in relation to ALSA (Advanced Linux Sound Architecture) 	5-2-3-II

		<ul style="list-style-type: none"> - OSS/3D - Digital voice device (linear encoding) - Mixer - Synthesizer - Music (mu-law Encoding) - MIDI (Interface with keyboards, synthesizers, stage props) 	
	<p>III</p>	<ul style="list-style-type: none"> - Audio Programming - Opening the Sound Device - Sampling Parameters - Audio Format - Channels - Sampling Rate - Half & Full Duplex - Input/Output - Multitasking 	<p>5-2-3-III</p>

SKILL CATEGORY NAME	Development System		SKILL CATEGORY NO.	6
SKILL NAME	Development Frameworks		SKILL NO.	1
TOPICS	LEVEL	SUBTOPICS		CODE NO.
Introduction of development framework				
categories and features of development frameworks				
OSS frameworks for Web applications				
The outline of free Web containers/J2EE containers				
Open source development tools				
Development process on deployment of frameworks				

SKILL CATEGORY NAME	Development System		SKILL CATEGORY NO.	6
SKILL NAME	Development Tools		SKILL NO.	2
TOPICS	LEVEL	SUBTOPICS		CODE NO.
Development flow and tools				
The outline of software development environments				
The outline of software application development in Linux development environments				
Version management tool utilization				
Program debugging environments using debuggers				
Debugging using kernel debuggers				
Debugging using bug tracking systems				
Types and functions of open source development tools				
Development procedures in integrated development environments				
Types and features of open source integrated development environments				
Workshops on software				

development in Linux development environments			
The outline of software development support tools in Linux development environments			
Software development tool evaluation			
Software development using Eclipse			

SKILL CATEGORY NAME	Development System		SKILL CATEGORY NO.	6
SKILL NAME	Integrated Development Environment (IDE)		SKILL NO.	3
TOPICS	LEVEL	SUBTOPICS		CODE NO.
<TBD>				

SKILL CATEGORY NAME	RDB		SKILL CATEGORY NO.	8
SKILL NAME	RDB System Management		SKILL NO.	2
TOPICS	LEVEL	SUBTOPICS		CODE NO.
<TDB>				

SKILL CATEGORY NAME	RDB		SKILL CATEGORY NO.	8
SKILL NAME	RDB Applications Development		SKILL NO.	3
TOPICS	LEVEL	SUBTOPICS		CODE NO.
Introduction to RDB application development	I	Objective	Understanding the basics of DB oriented applications such as Data Warehouse and Business Intelligence.	8-3-1-I
		Prerequisite		
		<ul style="list-style-type: none"> - RDB application overview - RDB application development methodology - RDB applications <ul style="list-style-type: none"> • Data Warehouse • Business Intelligence • Report Applications • Analyze application logs 		
DB system architecture	I	Objective		8-3-2-I
		Prerequisite		
		<ul style="list-style-type: none"> - Client/Server database system architecture - Web database system architecture 		
RDB design and creation	I	Objective		8-3-3-I
		Prerequisite		
	<ul style="list-style-type: none"> - RDB design overview - RDB design methodology 			
	II	<ul style="list-style-type: none"> - ER modeling - Relational DB design and normalization - Workshop on database design - Database creation 		8-3-3-II
Writing queries with SQL and optimizing them	I	Objective		8-3-4-I
		Prerequisite		
		<ul style="list-style-type: none"> - SQL overview and syntax 		

		- Writing various queries with SQL	
	II	Objective	
		Prerequisite	
		- Optimization of SQL statements	
			8-3-4-II
Database programming with C/C++	II	Objective	
		Prerequisite	
		<ul style="list-style-type: none"> - Introduction to DB programming with C/C++ - MySQL C/C++ API overview - Connecting to database - Querying database - Retrieving the query results - Inserting, deleting, and updating data in database - Using prepared statement - <u>Handling of Multiple Statement Execution</u> - Using transaction - Disconnecting from database 	
			8-3-5-II
	III	Objective	
Prerequisite			
- Making <u>a threaded client</u>			
			8-3-5-III
DB programming with JDBC and JSP/Servlet	II	<ul style="list-style-type: none"> - JDBC and/or JSP/Servlet programming overview - JDBC driver types - Installing and configuring JDBC and TOMCAT environment <ul style="list-style-type: none"> • JDBC interface overview • Setting up a connection • Querying database • Retrieving the query results 	

		<ul style="list-style-type: none"> • Inserting, deleting, and updating data in database • Using transaction • Using cookie and session • Disconnecting from database • JDBC exception handling • Using prepared statements and stored procedure • O/RM (Object-relational mapping) • Samples of DB application written using JDBC and JSP/Servlet • Practices in developing RDB applications using JDBC and JSP/Servlet 	
DB programming with PHP	II	Objective	
		Prerequisite	
		<ul style="list-style-type: none"> - Overview of PHP programming - Installing and configuring PHP and Apache - Setting up a connection - Querying database - Retrieving the query results - Inserting, deleting, and updating data in database - Using transaction - Disconnecting from database - Using prepared statements and stored procedures - Samples of Web DB application written by PHP - Practices in developing RDB applications using PHP 	
			8-3-7-II
Development of XML DB applications	II	Objective	
		Prerequisite	
		<ul style="list-style-type: none"> - Introduction to XML DB applications overview - XML and XML related tools 	
			8-3-8-II

	III	Objective		8-3-8-
		Prerequisite		
		<ul style="list-style-type: none"> - Mapping between tables in RDB and elements in XML - Relationships in RDB and containment & pointers in XML - Modeling exercise with example - Migrating an existing RDB to XML - Migrating existing XML documents to RDB - Querying RDB and writing XML by adding element tags yourself - Querying RDB and writing XML by using a utility module - Extracting records from XML with XML-parser and storing them into RDB - Extracting records from XML with XML-XPath and storing them into RDB - Samples of XML DB applications - Practices in developing XML DB applications 		

SKILL CATEGORY NAME	Testing		SKILL CATEGORY NO.	6
SKILL NAME	Software Testing		SKILL NO.	4
TOPICS	LEVEL	SUBTOPICS	CODE NO.	
Introduction to Software Testing	I	<ul style="list-style-type: none"> - Concept of testing - Verification and validation - Testing types - Principles of testing - Process of software testing 	6-4-1- I	
White-box Testing		<ul style="list-style-type: none"> - The process of white-box testing - Basis path testing - Structure-based testing <ul style="list-style-type: none"> • Condition testing • Data flow testing • Loop testing - Code review <ul style="list-style-type: none"> • Inspection • Desk check • Walkthrough 	6-4-2	
Black-box Testing		<ul style="list-style-type: none"> - Graph-based testing - Equivalent partitioning - Cause-effect analysis - Orthogonal array testing - Instrumentation <ul style="list-style-type: none"> • Testing oracle • Random data generator 	6-4-3	

Testing Coverage Analysis		<ul style="list-style-type: none"> - Code coverage analysis - Control flow coverage <ul style="list-style-type: none"> • Statement coverage • Decision coverage • Condition coverage • Condition/Decision coverage • Multiple condition coverage • Modified condition/Decision coverage • Path coverage - Data flow coverage <ul style="list-style-type: none"> • Rapps and Weyuker criteria • Ntafos criteria • Ural criteria • Laski and Korel criteria - Other criteria <ul style="list-style-type: none"> • Data domain coverage • Statistic or reliability coverage • Risk coverage • Safety coverage • State-based coverage 	6-4-4
Unit and Integration Testing		<ul style="list-style-type: none"> - Unit testing <ul style="list-style-type: none"> • Unit testing procedure • Unit testing limitation - Integration testing <ul style="list-style-type: none"> • Top-down integration • Bottom-up integration • Mixed integration • End-to-end integration 	6-4-5

Integration Testing	I	<ul style="list-style-type: none"> - Use of JUnit - JUnit design - Mock object <ul style="list-style-type: none"> • Boot virus • File virus • Micro virus • Mutation virus 	6-4-6- I
Regression Testing	I	<ul style="list-style-type: none"> - Characteristics of regression - Regression process - Regression strategies - Ripple effect analysis <ul style="list-style-type: none"> • The steps of ripple effect analysis • Program slices - Regression cost analysis 	6-4-7- I
State-based Software Testing	II	<ul style="list-style-type: none"> - State transition diagram (STD) - Statechart <ul style="list-style-type: none"> • Harel statechart • From statechart to STD • UML statechart - State-based testing <ul style="list-style-type: none"> • Process of State-based testing • State-based test case generation • Coverage analysis 	6-4-8- II
Object-oriented Application Testing	II	<ul style="list-style-type: none"> - OO testing approach <ul style="list-style-type: none"> • The availability of classic testing techniques • Fault-based testing 	6-4-9- II

		<ul style="list-style-type: none"> • The impact of OO techniques on testing • Test cases and class hierarchy <ul style="list-style-type: none"> - Class level testing - Inter-class testing 	
Web-based Application Testing	II	<ul style="list-style-type: none"> - The concepts of Web-based application testing - Testing process - Content testing - Interface testing - Component testing - Navigation testing - Configuration testing - Safety testing - Performance testing 	6-4-10- II
Testing Tools		<ul style="list-style-type: none"> - Introduction to testing tools - Parasoft- C++ Test - Coding Standards Static Analysis - Data Flow Static Analysis <ul style="list-style-type: none"> • Using data flow Static Analysis • Reviewing data flow Static Analysis Results • Customizing data flow Static Analysis - Code Review <ul style="list-style-type: none"> • Code Review Introduction • Code Review Scanner Configuration and Execution • Understanding the Code Review UI • Reviewers - Reviewing Code Modifications 	6-4-11

		<ul style="list-style-type: none"> • Authors - Examining and Responding to Review Comments • Monitors - Overseeing the Review Process • Code Review Tips and Tricks <p>- Test Generation and Execution</p>	
--	--	---	--

* Pre-requisite: Software engineering, C, C++, and Java programming languages.

* Objectives: Study of conventional software testing techniques including white-box testing, black-box testing, unit testing, integration testing and regression testing, study of techniques for testing object-oriented systems and Web applications, application a software testing tools.