

Annexure 'A'

Rajarshi Shahu Mahavidyalaya, Latur (Autonomous)

Department of Computer Science and IT

Curriculum Structure with effect from June, 2014

B. Sc. II (Computer Science) Semester III

| Course Code | Title of the course with paper number | Hours/ Week | Marks (50) | | Credits |
|----------------------|--|----------------|------------|-----------------|-----------|
| | | | Internal | End Semester | |
| U-COS-343 | Paper-V Computer Networks | 03 | 20 | 30 | 02 |
| U-COS-344 | Paper-VI Software Engineering | 03 | 20 | 30 | 02 |
| U-COS-345 | Laboratory Course -III | 03 | 20 | 30 | 01 |
| U-COS-346 | Laboratory Course -IV | 03 | 20 | 30 | 01 |
| Total Credits | | | | | 06 |

Student Stay Hours: 12/Week

B. Sc. II (Computer Science) Semester IV

| Course Code | Title of the course with paper number | Hours/ Week | Marks (50) | | Credits |
|----------------------|--|----------------|------------|-----------------|-----------|
| | | | Internal | End Semester | |
| U-COS-443 | Paper-VII Programming in JAVA | 03 | 20 | 30 | 02 |
| U-COS-444 | Paper-VIII Linux Operating System | 03 | 20 | 30 | 02 |
| U-COS-445 | Laboratory Course-V | 03 | 20 | 30 | 01 |
| U-COS-446 | Laboratory Course-VI | 03 | 20 | 30 | 01 |
| Total Credits | | | | | 06 |

Student Stay Hours: 12/Week

B. Sc. – II [Computer Science] Semester III

Course: Computer Network

Course Code: U-COS-343

Learning Objectives:

- Build an understanding of the fundamental concepts of computer networking.
- Familiarize the student with the basic categorization and terminology of the computer networking area.
- Introduce the student to advanced networking concepts, preparing the student for entry Advanced courses in computer networking.

Course Outcomes:

After completing this course the student must demonstrate the knowledge and ability to:

- Independently understand basic computer network technology.
- Understand and explain Data Communications System and its components.
- Identify the different types of network topologies and protocols.
- Identify the different types of network devices and their functions within a network
- Understand and building the skills of sub netting and routing mechanisms.

UNIT I : Overview networking terminology

- 1.1. Network types- Transmission Media [10hrs]
- 1.2. Control Schemes Layered Architecture OSI Reference Model
- 1.3. TCP/IP Reference Model

UNIT II: Telephone Networks Leased Lines PSTN ISDN Broadband

- 2.1. Encapsulation [12hrs]
- 2.2 End-to-end argument.
- 2.3 Protocol design issues.
- 2.4 Naming and Internet addressing, subnets.
- 2.5 DNS
- 2.6. Direct Link Networks, encodings, modulation.
- 2.7 Error detection and correction, CRC, Internet Checksum
- 2.8 CSMA/CD, Ethernet, Addressing

UNIT III : Bridges, Spanning Tree, Flooding/Multicasting

- [13hrs]
- 3.1. Switching vs. routing,
- 3.2 switching architectures
- 3.3. ARP, IP
- 3.4 The best effort service model, IP header structure.
- 3.5 Transport introduction, Port numbers, service models,
- 3.6 UDP Intro to reliability

UNIT IV: Introduction to TCP

[10hrs]

- 4.1. Flow control vs. congestion control
- 4.2 Congestion collapse
- 4.3 Window-based and rate-based congestion control
- 4.4 Wireless TCP DNS
- 4.5 Electronic Mail
- 4.6 FTP TFTP SNMP etc.

References/Books:

1. Computer Networks (Andrew S Tanenbaum)
2. Data and Computer Communications (William Stallings)

B. Sc. – II [Computer Science] Semester III

Course: Software Engineering

Course Code: U-COS-344

LERNING OBJECTIVES:

- Knowledge of basic software engineering methods and practices, and their appropriate application.
- Describe software engineering layered technology and Process frame work.
- A general understanding of software process models such as the waterfall and evolutionary models.
- Understanding of the role of project management including planning, scheduling, risk management, etc.
- Understanding of different software architectural styles.
- Understanding of approaches to verification and validation including static analysis, and reviews.
- Understanding of software testing approaches such as unit testing and integration testing.

COURSE OUTCOMES:

- Basic knowledge and understanding of the analysis and design of complex systems.
- Ability to apply software engineering principles and techniques.
- Ability to develop, maintain and evaluate software systems.
- Ability to perform independent research and analysis.
- Ability to work as an effective member or leader of software engineering teams.
- Ability to understand and meet ethical standard s and legal responsibilities

UNIT I: Software Engineering Fundamentals and Software Process

- 1.1 Definition of Software [13hrs]
- 1.2 Software characteristics
- 1.3 Software Applications
- 1.4 Software Process Models
 - 1.4.1 Waterfall model
 - 1.4.2 Prototyping model
 - 1.4.3Spiral model
 - 1.4.4 Incremental model
 - 1.4.5 Concurrent development model
- 1.5 Project management Concepts:
- 1.6The Management Spectrum - The People, The Product, The Process, and The Project.

UNIT II : Software Process and Project Metrics:

- 2.1 Measures, Metrics and Indicators, [10hrs]

- 2.2 Software measurement: Size - Oriented Metrics, Function - Oriented Metrics, Extended Function point metrics
- 2.3 Software Project Planning
- 2.4 Project Planning Objectives
- 2.5 Software Project Estimation
- 2.6 Decomposition Techniques - Problem Based Estimation, Process Based Estimation
- 2.7 Empirical Estimation Models- The COCOMO Model

UNIT III : Risk Analysis and Management and Software Quality Assurance

- 3.1 Software risks [12hrs]
- 3.2 Risk identification
- 3.3 Risk Projection
- 3.4 Risk Refinement
- 3.5 Risk Mitigation, Monitoring and Management
- 3.6 Basic concepts- Quality, Quality Control, Quality Assurance, Cost of Quality
- 3.7 Software Quality Assurance (SQA)
- 3.8 Formal Technical Review

UNIT IV: Coding and Testing

[10hrs]

- 4.1 Programming principles and guidelines
- 4.2 Incrementally developing code
- 4.3 Managing, evolving code
- 4.4 Unit testing
- 4.5 Code inspection
- 4.6 Testing concepts
- 4.7 Testing process
- 4.8 Black-box testing
- 4.9 White-box testing

Recommended Books:

1. R. Pressman: Software Engineering, McGraw-Hill.
2. K.K. Agrawal and Y. Sing: Software Engineering, New Age International.
3. P. Jalote: Software Project Management in Practice, Pearson.

B. Sc. – II [Computer Science] Semester III

Course: Lab Course III

Course Code: U-COS-345

Learning Objectives:

- Build an understanding of the fundamental concepts of computer networking.
- Familiarize the student with the basic categorization and terminology of the computer networking area.
- Understanding the procedure to create and manage LANs.

Course Outcomes:

- After completing this course the student must demonstrate the knowledge and ability to:
- Independently understand basic computer network technology.
- Create and Manage Local Area Networks
- Identify the different types of network topologies and protocols.
- Identify the different types of network devices and their functions within a network
- Understand and building the skills of sub netting and routing mechanisms.

Practical List

1. Study of different types of Network cables.
2. Practically implement the cross-wired cable and straight through cable using clamping tool.
3. Study of Network Devices in Detail.
4. Test TCP/IP Setup and create user accounts and user group.
5. Study of network IP.
6. Connect the computers in Local Area Network.
7. Study of basic network command and Network configuration commands.
8. Configure a Network topology using packet tracer software.
9. Configure a Network using Distance Vector Routing protocol.

B. Sc. – II [Computer Science] Semester III

Course: Lab Course IV

Course Code: U-COS-346

LEARNING OBJECTIVES:

- Knowledge of basic software engineering methods and practices, and their appropriate application.
- General understanding of software process models such as the waterfall and evolutionary models.
- Understand the various approaches to verification and validation including static analysis, and reviews.

COURSE OUTCOMES:

- Student will be able to collect the information for project development according to the development process
- Student will be able to develop small project.
- Student will be able to develop small project.
- Student will be able to test the projects.

Practical List

1. Study the complete Software Development Life Cycle (SDLC) and analyze various activities conducted as a part of various phases. For each SDLC phase, identify the objectives and summaries outcomes.
2. Consider any project to be developed in any technology as a Software Architect or Project Manager.
3. Construct Software Requirement Specification (SRS) document for the project.
4. Design the project.
5. Modeling of the project.
6. Study of prototyping software development model.
7. Study of incremental Software development Model and use it for incremental version of project.
8. Study of database generation for the project.
9. Creation of reports of the project.
10. Study of different testing techniques for the software.

B. Sc. – II [Computer Science] Semester IV

Course: Programming in JAVA

Course Code: U-COS-443

LEARNING OBJECTIVES:

- To develop programming ability in the Java programming language.
- Knowledge of object-oriented paradigm in the Java programming language.
- The use of Java in a variety of technologies and on different platforms.

COURSE OUTCOMES:

- Ability to understand the structure and model of the Java programming language.
- Students will be able to use the Java programming language for various programming technologies.
- Ability to develop software, application in the Java programming language.

UNIT- I: An Introduction to Java

- 1.1.1 A Short History of Java [12hrs]
- 1.1.2 Features of Java
- 1.1.3 Difference between Java and C++
- 1.1.4 Java virtual machine (JVM)
- 1.1.5 Java program structure
- 1.1.6 Java statement
- 1.1.7 Types of Comments
- 1.1.8 Keywords, Data Types
- 1.1.9 Variables and Constants
- 1.1.10 Operators
- 1.1.11 Output using println() method
- 1.1.12 Simple java program
- 1.1.13 Command Line Arguments

UNIT – II: Decision Making, Branching, Looping and Classes, Object and Methods

- 2.1 Decision making with if statement [13hrs]
- 2.2 Simple if statement, if...else statement, Nesting of if...else
- 2.3 Switch statement
- 2.4 while statement, do statement, for statement.
- 2.5 Introduction of classes
- 2.6 Defining a class, Adding variables, Adding Methods, Accessing Class Members, 2.7 Constructors
- 2.8 Method Overloading
- 2.9 Static Members
- 2.10 Inheritance: Extending a class, Overriding Method

UNIT –III: Arrays, Strings, Vectors and Creating and Using Packages

- 3.1 Introduction [10hrs]
- 3.2 One-dimensional Arrays, Creating an one dimensional array
- 3.3 Two-dimensional Arrays, Creating an two dimensional array
- 3.4 String Arrays
- 3.5 String Methods
- 3.6 Java API package
- 3.7 Using system packages
- 3.8 Naming Conventions
- 3.9 Creating Packages
- 3.10 Accessing a package
- 3.11 Using a Package
- 3.12 Adding a class to a package.

UNIT – IV: Exception Handling and Applet Programming

- 4.1 Dealing Errors [10hrs]
- 4.2 Catching exception and exception handling
- 4.3 Creating user defined exception.
- 4.4 Applet Life Cycle
- 4.5 Applet HTML Tags
- 4.6 Passing parameters to Applet
- 4.7 Repaint() and Update() method

Reference Books :

Programming with Java , A primer ,Forth edition , By E. Balagurusamy
Complete reference Java by Herbert Schildt(5th edition)
Java 2 programming black books, Steven Horlzner
Java Servlets Programming by Jason Hunter, O'Reilly

B. Sc. – II [Computer Science] Semester IV

Course: Linux Operating System

Course Code: U-COS-444

COURSE OBJECTIVES:

Skills and knowledge students will gain in this course:

- Install Red Hat based Linux distributions;
- Maintain operating system updates;
- Understanding of each of the following:
- Basics of Linux security;
- Major issues involved in Open Source licensing;
- Basics of Linux scripting and be able to write a simple script;

COURSE OUTCOMES:

- After successfully completing this course, students should be able to –
- Comfortably use basic Linux commands from the command line (from a terminal window);
- Organize and manage their files within the Linux file system;
- Usefully combine UNIX/Linux tools using features such as filters, pipes, redirection, and regular expressions;
- Customize their UNIX/Linux working environment;
- Be knowledgeable enough about basic Linux shell scripting to be able to successfully read and write shell scripts;

UNIT I: Introduction and Working with permissions

- 1.1.1 History of Linux [12hrs]
- 1.1.2 Directory structure of Linux
- 1.1.3 Comparison of various operating systems
- 1.1.4 Advantages of Linux, Flavors of Linux, Installation notes, Linux Loader, Linux kernel
- 1.1.5** Linux file system – ext3, ext2
- 1.1.6 Assigning file permission
- 1.1.7 Directory Permission
- 1.1.8 Using text editors, Working with vi editor
- 1.1.9 System services and run levels

UNIT II : Linux commands

- 2.1 General Purpose Utilities – Man, who, login, logout, shutdown, passwd. [10hrs]
- 2.2 File management commands – cat, cp, rm, mv, wc,

- 2.3 Filter related commands – grep ,wc, head, tail, sort
- 2.4 Directory management commands – pwd, cd, mkdir, rmdir, ls
- 2.5 Checking directories and permissions – pwd, chmod, cd, ls
- 2.6 Compression commands zip ,unzip, gzip, gunzip, printer related commands lpc ,lpr,lpq.lprm

UNIT III: Backup strategies and System Administration

- 3.1 Choosing Backup Strategies and Operations [13hrs]
- 3.2 Choosing Backup hardware and media.
- 3.3 Using backup software and commands
- 3.4 Performing system maintenance
- 3.5 Communication commands:- write, wall, talk, mesg, motd, Pre-login Message
- 3.6 Managing software with RPM- Installing, Uninstalling, Upgrading
- 3.7 Managing users and managing Groups and managing passwords.

UNIT IV: Network configuration for Linux

- 4.1 Network configuration tools [8hrs]
- 4.2 Dynamic host configuration protocol.
- 4.3 Network files system.

References Books:

Bill Ball, David Pitts, “Red Hat Linux 7 Unleashed”, Techmedia SAMS Publication
EviNemeth, Garth Snyder, Scott Seebass, Trent R. Hein, “UNIX System Administration Handbook” Person Education Asia (LPE) (IIIrd Edition)
Red hat Linux & fedora unlashed Authors Bill Ball & Hoyt Dust.

B. Sc. – II [Computer Science] Semester IV

Course: Lab Course V

Course Code: U-COS-445

COURSE OBJECTIVES:

- To develop programming ability in the Java programming language
- Knowledge of object-oriented paradigm in the Java programming language
- The use of Java in a variety of technologies and on different platforms

COURSE OUTCOMES:

- Ability to understand the structure and model of the Java programming language
- Students will be able to use the Java programming language for various programming technologies.
- Ability to develop software, application in the Java programming language

Practical (JAVA Programming)

1. Installation of JAVA
2. Simple JAVA Programs
3. Programs to demonstrate static members
4. Programs to demonstrate method overriding
5. Programs to demonstrate constructor
6. Programs to demonstrate Inheritance in JAVA
7. Programs to demonstrate method overriding
8. Programs to demonstrate array in JAVA
9. Programs to demonstrate string functions
10. Programs to create package in JAVA
11. Programs to demonstrate Threads in JAVA

B. Sc. – II [Computer Science] Semester IV

Course: Lab Course VI

Course Code: U-COS-446

COURSE OBJECTIVES:

- Install Red Hat based Linux distributions;
- Maintain operating system updates;
- Study of general purpose utilities commands.
- Study of VI editor
- Understanding basics of Linux security.

COURSE OUTCOMES:

- After successfully completing this course, students should be able to –
- Comfortably use basic Linux commands from the command line (from a terminal window);
- Organize and manage their files within the Linux file system;
- Usefully combine UNIX/Linux tools using features such as filters, pipes, redirection, and regular expressions;

Practical List

1. Introduction to LINUX Operating System.
2. Installation of LINUX Operating System (Red Hat-5).
3. Study of general purpose utilities commands.
4. Study of user & session management commands.
5. Study of file system navigation commands, text processing tools, communication commands.
6. Study of VI editor.
7. Study of Shell Script.
8. C programs in Linux.
9. Installation using RPM/YUM server.
10. Back up using TAR command.