ANNA UNIVERSITY, CHENNAI – 25
CENTRE FOR DISTANCE EDUCATION

MASTER OF SCIENCE - M.Sc. (CS-FOSS) – ONLINE MODE

CURRICULUM - 2013

SEMESTER - I

Code No.	Course Title	Credits
DIF 7101	Introduction to Computing	3
DIF 7102	Philosophy and practice of FOSS	3
DIF 7103	Basics of GNU/Linux Operating System & FOSS tools (with Lab component)	5
DIF 7104	Object Oriented Programming with Data Structures (with Lab component)	5
DIF 7105	HTML, CSS, and Javascript (with Lab component)	5
	TOTAL	**21**

SEMESTER - II

Code No.	Course Title	Credits
DIF 7201	Introduction to Java Technology (with Lab component)	5
DIF 7202	Software Development Practices (with Lab component)	5
DIF 7203	RDBMS Basics (with Lab component)	5
DIF 7204	Computer Networks and the Internet	3
DIF 7205	Business Communications in English	3
	TOTAL	**21**

SEMESTER - III

Code No.	Course Title	Credits
DIF 7301	Application Security	3
DIF 7302	Web Development (with Lab component)	5
DIF 7303	Application Development using PHP (with Lab component)	5
DIF 7304	Mobile Application Development (with Lab component)	5
DIF 7305	Collaborative development with Version Control	3
	TOTAL	**21**

SEMESTER - IV

Code No.	Course Title	Credits
E1	Elective – I	3
E2	Elective – II	3
DIF 7411	Project Work	12
	TOTAL	**18**

LIST OF ELECTIVES

Code No.	Course Title	Credits
DIF 7001	Virtualisation and Cloud Computing	3
DIF 7002	Realtime and Embedded Programming	3
DIF 7003	Software Testing	3
DIF 7004	Python programming	3

ANNA UNIVERSITY
M.Sc. (CS-FOSS) – ONLINE MODE
REGULATIONS – 2013
SYLLABUS I TO IV SEMESTERS

SEMESTER – I

DIF 7101 INTRODUCTION TO COMPUTING Credits : 3

COURSE OBJECTIVES

- Understand the evolution, architecture and devices of digital computers
- Understand the functions and types of software used in digital computers
- Get an insight into Programming Languages with some examples
- Become aware of the concepts and types of Computer networks and the Internet

COURSE OUTCOMES

- An understanding and appreciation of the evolution of computing
- Awareness of current status of the Hardware and Software of computers
- Exposure to computer programming
- Understanding of network of computers

UNIT I History of Computers, Capabilities and Limitations, Types of Computers, Generations of Computers – past, present and future, configuration and evolution – 16, 32, 64 bit systems – graphics – multimedia – networks.

UNIT II Von Neumann Architecture, Basic components of a computer – CPU – Control unit, ALU, Registers, I/O units -- , Memory – Cache, RAM, ROM, PROM, EPROM --, I/O Device – keyboard, mouse, trackball, display, blue-tooth, firewire, USB, Storage Devices – internal, external, Multimedia Devices – CDROMS, DVD, Blu-Ray drives, sound cards, graphics cards.

UNIT III Software and its need, Types of software – System Software – OS, Device Drivers, Assemblers, Compilers, Interpreters, utilities -- Application Software – Office suites – word processors, spreadsheets, presentation software, DBMS – graphics software – browsers – personal desktop applications (accounting, planner).

UNIT IV Programming Languages – algorithms – flowcharts – machine language, assembly, High level, 4GL, Scripting languages (shell, awk, Perl, Python) -- Types – procedural (FORTRAN, BASIC, COBOL, Pascal, C) , object oriented (C++, Java, Javascript), functional (Lisp) etc.

UNIT V Computer networking, protocols, channels (Copper, fiber, wireless etc), types (LAN, WAN etc.), networking components (NIC, Routers, Switch, Hub, etc), topologies (ring, bus, star, mesh, tree), the Internet (evolution, protocols, www).

TEXT BOOKS:

1. Nell Dale and John Lewis, “Computer Science Illuminated”, Narosa Publishing House, Fifth Edition, 2012
2. S. Jaiswal, “Fundamentals Of Information Technology”, Galgotia Publication P Ltd, 2003

REFERENCE BOOKS:

1. http://en.wikiversity.org/wiki/Introduction_to_Computers

DIF 7102 PHILOSOPHY AND PRACTICE OF FOSS Credits : 3

COURSE OBJECTIVES

- Understand the concept and evolution of FOSS with examples
- Realise the significance of FOSS Communities and the FOSS development process
- Become aware of FOSS licenses and the processes of FOSS adoption and use
- Get exposed to the idea of Open Standards and practices

COURSE OUTCOMES

- Understanding and appreciation of the FOSS Movement
- Knowledge of FOSS Communities, FOSS Development Model and FOSS Licenses
- Understanding the issues in FOSS adoption

UNIT I Definition of open source software, essential requirements for being open, free software vs open source software, 4 degrees of freedom wrt software, notion of a fork, popular FOSS examples (firefox, Linux, Apache, Moodle, openoffice).

UNIT II Notion of community, guidelines for effectively working with FOSS community, benefits of community based software development, FOSS licensing models, copyleft movement, popular FOSS licenses (GPL, LGPL, BSD, etc.), implications of FOSS license for developer and user.

UNIT III FOSS development process and basic principles. Quality control process, different categories of players in FOSS development, need for frequent release, many-eyeball effect, development tools: IDE, languages, toolkits, testing tools, etc.

UNIT IV FOSS adoption: governments and FOSS, case studies of government interventions and policies, FOSS in educational context, major industries adopting FOSS, migrating to FOSS: processes and challenges, FOSS in India.

UNIT V Notion and importance of open standards, some popular open standards examples, notion of open content, examples of open content, creative commons and other popular open content license norms.

TEXT BOOK:

1. Jesús M. González-Barahona et al, “Introduction to Free Software”, Free Technology Academy, Europe, 2009.

REFERENCE BOOKS:

1. **http://ftacademy.org/materials/fsm/1#1**
2. Online course at **http://202.141.152.108/nrcfoss**

DIF 7103 BASICS OF THE GNU/LINUX OPERATING SYSTEM & FOSS TOOLS (WITH LAB COMPONENT) Credits : 5

COURSE OBJECTIVES

- Understanding the principles of GNU/Linux and distributions
- Installation and configuring Linux and using shell commands
- Understanding the basic services and advanced tasks in Linux
- Acquiring hands on exposure to the above topics

COURSE OUTCOMES

- Ability to Install, Configure and Program in Linux
- Provide and manage services in Linux
- Ability to Network using Linux with security

UNIT I Introduction to GNU/Linux and basics: What is Linux? Linux distributions -- File System Introduction -- File System Hierarchies-- File system security with reference to SELinux --Types of partitions and mount options -- RAID overview-- Logical Volume Management.

UNIT II Installation and Configurations: Types of Installation Media and examples -- Installation of GNU/Linux -- Start Up and Shut Down Sequences -- Run Levels-- Basic System Navigation-- Window manager, Desktop environments – KDE, GNOME -- Editors (vim, nano, emacs and GUI editors).

UNIT III Using GNU/Linux (Shell Commands): Overview of Shells – BASH and other important shells -- Process Management -- User Administration-- Networking overview and commands --Introduction to IO redirection-- Software Repositories.

UNIT IV Services in Linux: Apache Web Server and PHP-Firewall services using IP tables-- Openssh -- CUPS printing system – Databases (MySQL) and basic administration.

UNIT V Advanced tasks in Linux : Installation of software using source packages, gcc compilation – Securing the system – Backups and file sharing with reference to rsync and LVM snapshots – GRUB and rescue mode operations – Log processing with reference to tools like awstat-- BASH programming-- SVN and basic operations.

LAB UNIT Installation-Configurations: Installation of Gnu/Linux , installing/removing/ upgrading SW packages using package management tools , Setup of network interface - Shell scripting/programming - Services in Linux: configurating of LAMP stack , Tomcat , CUPS , Firewal (openssh) , Crontab , Samba configuration - Advanced tasks in linux: SW Installation using source packages , GCC compilation , build process (make, automake, autoconf) - Securing the system: Backups and file sharing, GRUB and rescue mode operations , Version Control System-- setup and usage of SVN.

TEXT BOOKS:

1. N. B. Venkateshwarlu (Ed), "Introduction to Linux Installation and Programming", B S Publishers, Hyderabad, 2005.(An NRCFOSS Publication)
2. Matt Welsh, Matthias Kalle Dalheimer, Terry Dawson, and Lar Kaufman, Running Linux, Fourth Edition, O'Reilly Publishers, 2002.
3. Carla Schroder, Linux Cookbook, First Edition, O'Reilly Cookbooks Series, 2004.

REFERENCE BOOKS:

1. Michael Stutz, “The Linux Cookbook: Tips and Techniques for Everyday Use”, First Edition, 2001. URL:http://dsl.org/cookbook/cookbook_toc.html
2. “The Linux System Administrator's” Guide, URL:http://tldp.org/LDP/sag/html/sag.html
3. The Software-RAID HOWTO, http://www.tldp.org/HOWTO/Software-RAID-HOWTO.html
4. Mendel Cooper, “Advanced Bash Scripting Guide”, June 2005. URL:http://www.tldp.org/guides.html
5. Olaf Kirch, “The Network Administrators” Guide, URL:http://tldp.org/LDP/nag/nag.html

DIF 7104

OBJECT ORIENTED PROGRAMMING WITH DATA STRUCTURES (WITH LAB COMPONENT)

Credits : 5

COURSE OBJECTIVES

- Understand the basics of Object Oriented Programming Concepts
- Understanding Objects and Abstractions
- Understanding Data types, Operators, Functions Memory management
- Comprehending the concept of Inheritance and its applications
- Learn the meaning and use of Templates and their roles
- Understand the basic Data Structures such as Linked List, Stacks and Queues

COURSE OUTCOMES

Upon Completion of the course, the students would be able to:

- Understand and work with Objects and Object oriented approaches and techniques
- Work with Inheritance concepts and functions in OOP
- Use Template based techniques in OOP
- Design, implement and debug C++ programs for Data Structures

UNIT I Introduction to Programming -Programming techniques-review of structures-Unstructured ,procedural, modular programming-Object Oriented Programming(OOPS)-Overview-concepts of OOPS-Data Structures-single lists, Multiple lists - Data Abstraction & Abstract data types -C++ programming on Linux platform -GCC& GDB.

UNIT II Introduction to objects -Making& using objects -Progress of Abstraction-Object and interface-Inheritance-reusing the interface-interchangeable objects with polymorphism-creating and destroying objects -Exception handling : dealing with errors.

UNIT III Data types, operators-creating functions – controlling execution-Classes-handling classes-Base class initialisation-Nested classes-Dynamic Memory management – memory allocation, de-allocation-Initialisation and clean up-constructors and destructors-stash & stacks.

UNIT IV Inheritance-Function overriding-protected Access specifiers-Types of Inheritance Virtual and Pure virtual functions-Function overloading-operator overloading-Inheritance and Composition- Polymorphism & virtual functions.

UNIT V Introduction to templates -overview of templates-template syntax-Stack & Stash as templates-Holding objects by values-Iterators-Stack with iterators – Linked List, Queues, Trees.

LAB UNIT Lab Classes would cover Creation and execution of simple programs using GCC&GDB – Instantiation – Inheritance with types – Polymorphism – Exception Handling – Use of Constructors & Destructors – Overloading with Types – Templates.

TEXT BOOKS:

1. Sourav Sahay, “Object-Oriented Programming with C++”, Oxford University Press, 2006.
2. Bruce Eckel, “Thinking in C++ - 2nd Edition, Pearson Education, First Edition, 2006
3. Mark Allen Weiss, “Data Structures and Algorithm Analysis in C++”, Pearson Education, 2nd edition, 2005

REFERENCE BOOKS:

1. Yashwant Kanetkar, “Let us C++”, BPB Publication, 13th Edition, 2012
2. James P Cohoon & Jack W Davidson, “Programming in C++”, Tata McGraw Hill, Special Indian Edition, 2007

DIF 7105 HTML, CSS AND JAVASCRIPT (WITH LAB COMPONENT) Credits : 5

COURSE OBJECTIVES

- Understanding HTML and Elements
- Learning CSS and its functions
- Learning Javascript and Advanced Javascript
- Acquire practical skills on the above concepts

COURSE OUTCOMES

- Ability to build HTML and CSS applications on the web, including HTML 5 programming
- Build web applications using advanced features in Javascript

UNIT I Introduction to World Wide Web-Basics of HTML - Introduction to mark-up - XHTML and HTML - Basic document structure - DTD - Tags and attributes - Inline text elements - Block and special elements.

UNIT II Hyper link Elements - Floating Elements - Frames and Framesets - Tables - Lists - Forms - Form elements - Handling special characters - Handling images - Multimedia elements.

UNIT III Basics of CSS - Elements of styles - Style syntax - Classes - Properties - Applying styles to document - Selectors - Dimensions / Grouping & Nesting - CSS Display & Positioning - Display & Alignments - Inheritance -Media types - Pseudo class - Sprites.

UNIT IV Introduction to JavaScript - Core features - Data types and Variables - Operators, Expressions, and Statements - Functions - Objects - Array, Date and Math related Objects - Document object Model - Event Handling - Controlling Windows & Frames and Documents - Form handling and validations - Dynamic effects with JavaScript - Iterative methods.

UNIT V Advanced JavaScript - Ajax - Browser deduction and using embedded objects - Using XML and JSON - Object oriented JavaScript - Object constructor and Prototyping - Sub classes and Super classes - Introduction to Javascript frameworks [Jquery, Dojo, Prototype etc].

LAB UNIT Lab Classes would cover – Basic HTML Programs covering wide range of tags & attributes – HTTP Methods – Tables – Frames & Framesets – Multimedia elements – Javascripts – CSS – Media Types – Sprites – Programs using HTML 5 – Form Handling – Dynamic Effects with Java Scripts.

TEXT BOOKS:

1. Thomas Powell, “HTML & CSS: The Complete Reference”, The McGraw Hill Companies, Fifth Edition
2. Thomas Powell (Author), Fritz Schen, “JavaScript: The Complete Reference”, Second Edition

SEMESTER – II

DIF 7201 INTRODUCTION TO JAVA TECHNOLOGY (WITH LAB COMPONENT) Credits : 5

COURSE OBJECTIVES

- Get an introduction to Java language
- Learn to program in Java
- Understand Advance Java and Enterprise Java
- Networking with Java

COURSE OUTCOMES

Upon Completion of the course, the students would be able to:

- Work with Java I/O streams, networking and GUI based application development
- Work with Web application development using Java Server Faces
- Work with Security features supported in Java
- Develop web services using REST/SOAP/JSON
- Design and develop applications using other frameworks

UNIT I **JAVA ESSENTIALS -** Introduction to Java, Java model, JVM, Setting up Java environment (downloading and installing J2SE/JDK/OpenJDK, Classpath Setting, Memory Setting, Eclipse for Java), Hello World in Java - Basics - program structure, variables, primitive types, operators & their precedence - Flow control - if-then, logical operators, switch, Loops - System package - Basic I/O - Using Java API Documentation.

UNIT II **GETTING INTO DEPTH** - OOP in Java - Class, Object, Inheritance, Interfaces, Packages. Class anatomy - declaring a class, defining methods, members, access modifiers. Abstract classes, Interfaces, overloading, inheritance, Nested classes, Inner classes, Enums - Objects - creating objects, instantiating classes, using constructors, referring members - Methods – declaration, scope, overloading - Packages – creating, naming, managing source files - Arrays - declaring, manipulating - Strings - Using Strings, StringBuffer classes - Generics - defining. Boxing/Unboxing.

UNIT III **ADVANCED JAVA -** Exceptions - throwing and catching exceptions, chaining - I/O Streams - types, Readers/Writers, formatting classes, File I/O - Multithreading - processes, threads, synchronization - Java.util.* - Java utils classes link Properties, command line arguments - Collections framework - Interfaces and Implementation class for Lists, Maps, Hashtable - Annotations & Reflection - Object Serialization - Design patterns in Java - Garbage Collections.

UNIT IV **GUI & INTERNATIONALIZATION, JDBC, NETWORKING -** Swing Toolkit – Swing classes , JComponent, Container classes, Layouts managers, Event listeners - Applets, Security model - Internationalization in Java – Locale, Locale specific data, Formatting, Localization using properties - JDBC – creating connections, SQL queries, RowSets, Prepared Statements, Stored procedures - Networking – Using URLs, sockets, datagrams.

UNIT V **ENTERPRISE JAVA -** J2EE architecture, Web services concepts, Application servers - Java Beans, Servlets, Apache Tomcat servlet container - Java Server Pages, Java Server Faces- Using Ant/Maven.

LAB UNIT Exercises from all the Units

TEXT BOOKS:

1. Paul Deitel and Harvey Daitel, "Java – How to program", Ninth Edition, PHI, 2012
2. Java Tutorial online, http://docs.oracle.com/javase/tutorial/
3. Java Tutorial ebook download, http://www.oracle.com/technetwork/java/javase/downloads/java-se-7-tutorial-2012-02-28-1536013.html

REFERENCE BOOKS:

1. Ivor Horton's, "Beginning Java: Java", Seventh Edition, 2011
2. Herbert Schildt, "Java: The Complete Reference" Seventh Edition, McGraw Hill Professional, 2006
3. E. Balaguruswamy, "Java", Fourth Edition, McGraw Hill Professional, 2010

DIF 7202 SOFTWARE DEVELOPMENT PRACTICES Credits: 5 (WITH LAB COMPONENT)

COURSE OBJECTIVES

- Awareness of SW Development in conventional practices and in FOSS practice
- Learn about Modeling and Modeling Languages
- Understand Agile development and FOSS development
- Learn techniques of SW Project Management

COURSE OUTCOMES

- Ability to build SW/IT systems following FOSS development methodology and FOSS tools
- Ability to Manage FOSS Projects

UNIT I **INTRODUCTION TO THE SOFTWARE DEVELOPMENT PROCESS -** Software process – Process and Project., Component of Software Process, Software Development Process Model – Waterfall, Prototyping, Iterative development, RUP, Timeboxing, Agile and Extreme. Project Management Process.

UNIT II **METHODOLOGY,MODELING AND UNIFIED MODELING LANGUAGE -** Object oriented methodologies – Rumbaugh, Booch, Jacobson et al ,The Unified Approach.Unified Modeling Language -- introduction, static vs dynamic models, UML static Modeling- class diagram, use-case diagram, UML Dynamic Modeling-interaction, sequence, collaboration, Statechart, Activity, Implementation, Component and Deployment. FOSS tools for UML – AgroUML / Umbrello.

UNIT III **PROGRESSIVE METHODOLOGIES –** Agile software development methods-eXtreme Programming, SCRUM, Crystal family, Feature Driven development, Rational Unified Process, Dynamic Systems Development Method, Adaptive Software Development, Open Source Software Development.

UNIT IV **FOSS DEVELOPMENT PRACTICES & TOOLS:** Using – Bug reporting – Fixing: patching – Co-development – Developing – **Tools:** Communication Channels - Mailing list, Instant Chat, Wiki, Forums. FOSS IDE - Eclipse. Revision Control Systems - SVN. Bug tracking system - Bugzilla. Database Tools: PostgresSQL - PgAdminIII. Testing Tools - Selenium.

UNIT V **PROJECT MANAGEMENT:** The concept of BDFL and decision making process - Management – Build & Release – Cathedral & Bazaar Models. Package Management - DEB and APT. Release management - Release Cycle and Daily development. Project Management Tools - Trac. Documentation Tools - Doxygen, Docbook.

LAB UNIT Practical exercises to implement the features covered in units II, IV and V

TEXT BOOKS:

1. Roger S. Pressman, “Software Engineering – A practitioner’s Approach”, Seventh Edition, Tata McGraw-Hill International Edition, 2009
2. AU-KBC Centre, “Software Development Practices”, Textbook compiled and edited from available material, 2013

REFERENCE BOOKS:

1. Pankaj Jalote, “An Integrated Approach to Software Engineering”, Third Edition, Springer, 2005
2. Ali Bahrami, “Object oriented systems development using the unified modeling language”, Tata McGraw-Hill Edition
3. Kent Beck and Cynthia Andreas, “Extreme Programming Explained: Embrace Change”, Addison-Wesley; 2nd edition, 2000
4. Karl Fogel, “Producing Open Source Software”, Fifth Edition, O’Reilly, 2005, URL: http://producingoss.com/en/index.html
5. Mathew B. Doar, “Practical Development Environments”, First Edition, by ISBN: 9788184040050 Shroff/ O' Reilly, 2005

DIF 7203 RDBMS BASICS Credits: 5
(WITH LAB COMPONENT)

COURSE OBJECTIVES

- Understand concept of DMBS and RDBMS
- Learn SQL
- Learn about Data base administration
- Get introduced to RDBMS Programming

COURSE OUTCOMES

- Ability to build systems using FOSS RBMS systems like Post-gres and MySQL
- Ability to Manage and Administer FOSS RDBMS packages

UNIT I **AN INTRODUCTION TO DBMS -** History of DBMS – Navigational, Relational, Object Oriented, XML and NoSQL Databases, Database Models – Hierarchical, Network, Relational, Multidimensional and Object Oriented models, Relational Database System Architecture, RDBMS – Postgresql and Sqlite.

UNIT II **RELATIONAL DATABASES -** Relational Database Design – ER models, ER diagrams and Database normalization, Locks, Concurrency-Control-MVCC, ANSI-SQL Overview, Data Definition, Basic structures, Basic Operations.

UNIT III **SQL -** ANSI-SQL Data Modification Statements, Null values, Aggregate Functions, Nested Sub-queries, SQL Joins – Inner and Outer Joins, Views – creation and updating, Transactions, Integrity Constraints – not null, unique, check and referential integrity, Functions and Procedures, Triggers.

UNIT IV **RDBMS ADMINISTRATION -** Server Setup, Server Configuration, Managing Databases, Backup and Restore, Database Monitoring, High Availability, Load, Balancing, Replication and concepts of Clustering.

UNIT V **PROGRAMMING WITH RDBMS -** Embedded SQL – Cursors and Dynamic SQL using PgSQL, Embedded SQL in C using ECPG, JDBC access to Postgresql, Embedded Database in C – Using the Sqlite C/C++ API.

LAB UNIT **USING POSTGRESQL AND SQLITE** Exercises on using ANSI-SQL, DB administration (using tools like PgAdmin), Schema features of PostgreSQL and programming – from Units II, III, IV and V.

TEXT BOOKS:

1. Abraham Silberschatz, Henry F. Korth, S. Sudarshan, “Database System Concepts”, Sixth Edition, McGraw Hill, 2011
2. Simon Riggs, Hannu Krosing, “PostgreSQL 9 Administration Cookbook: Lite“, ISBN: 9789350232866 ; Shroff/ Packt Publishing Ltd, 2011

REFERENCE BOOKS:

1. Postgresql Manuals – http://www.postgresql.org/docs/
2. SQLite Documents – http://www.sqlite.org/docs.html
3. Planet PostgreSQL - http://planet.postgresql.org/

DIF 7204 COMPUTER NETWORKS AND THE INTERNET Credits: 3

COURSE OBJECTIVES

- Know network Fundamentals
- Understand the concepts of Networking & Internet
- Learn Networking with GNU/LIUX
- Understand Networking with FOSS
- Get introduced to Network Security and Management issues in FOSS

COURSE OUTCOMES

Upon Completion of the course, the students would be able to:

- Networking of GNU/Linux systems using FOSS tools and provide the required services
- Deploy and Manage networks like VPN and VLAN along with required security provisions

UNIT I **NETWORKING BASICS -** Network architecture – ISO/OSI Layers – Network Topologies, Medium Access Control techniques, TCP/UDP Protocols, Application layer services (Email, HTTP, FTP,DNS).

UNIT II **NETWORKING IN GNU/LINUX -** Issues in TCP/IP Networking, Comparison of IPV4 & IPV6, Configuration of TCP/IP Networking - IP configuration, hostname, gateway - Important Networking Commands - Dynamic Host Configuration Protocol (DHCP) and Preboot Execution Environment (PXE).

UNIT III **FOSS NETWORKING SERVICES -** Domain Name System (DNS), Simple Mail Transfer Protocol (SMTP), Post Office Protocol (POP3) and Internet Message Access Protocol (IMAP) LAPP - Apache Web-Server - virtual-hosting, SSL setup, performance tuning, PHP - configuration, PostgreSQL Databases - Setup and configuration.

UNIT IV **NETWORK SECURITY AND GNU/LINUX -** Firewall - , DMZ, proxy - System & Network Security best practices - Intrusion Detection Systems (IDS)- NMAP, SNORT - Application firewall - Apache mod_security, setup and configuration - Network Address Translation (NAT), Virtual Private Networks (VPN) and VLAN.

UNIT V **NETWORK MANAGEMENT IN GNU/LINUX -** Squid – configuration, caching, filtering and accelerator mode setup - Load-balancing with Apache, Case Study - Network Deployments - Campus and office networks - System/Network Monitoring - MRTG, Nagios/Zabbix/OpenNMS.

TEXT BOOKS:

1. Ying-Dar Lin, Ren-Hung Hwang, and Fred Baker, “Computer Networks: An Open Source Approach “,McGraw-Hill Education, 2011 - E-Book - http://oreilly.com/openbook/linag2/book/index.html
2. Free/Open Source Software Network Infrastructure and Security, Gaurab Raj IOSN

REFERENCE BOOKS:

1. http://portal.aauj.edu/portal_resources/downloads/operating_system/linux_security.pdf
2. Andrew S. Tanenbaum, “Computer Networks”, Fourth Edition, 2003

DIF 7205 BUSINESS COMMUNICATIONS IN ENGLISH Credits: 3

COURSE OBJECTIVES

- Understand the importance of proper communication in engineering practice
- Acquire Listening and Speaking skills in English
- Acquire reading and Writing skills in English
- Comprehend communication in the social and in the Business context

COURSE OUTCOMES

- Able to comprehend English through listening and reading, as well as express oneself clearly in English through speaking and writing
- Effective and competent in English Communication in social and business contexts

UNIT I **INTRODUCTION TO BUSINESS COMMUNICATION -** Definition and purpose of communication (why study communication; uniqueness of human communication; purpose of communication, etc.); types of communication (internal & external; functional, situational, verbal and non-verbal, interpersonal, group, interactive, public role of communication); communication across cultures; barriers to communication (wrong choice of medium, physical barriers, semantic barriers, comprehension of reality, socio-psychological barriers).

UNIT II **LANGUAGE SKILLS (LISTENING AND SPEAKING) -** Introduction to language skills (LSRW); Listening – (listening & hearing, active vs passive listening, etc.); barriers to effective listening; improving listening skills. Speaking - Speech vs enunciation; qualities of a good listener and a good speaker; feedback. Oral communication (meetings, telephone conversations, teleconferencing, web chat presentations, video or audio recordings); role play; group discussion.

UNIT III **LANGUAGE SKILLS (READING AND WRITING) -** Reading – Focus on the structure not on the theme alone; reading skills including sub-skills of reading (skimming & scanning); Writing – Precise language, tone, ordering, etiquette; Written communication (qualities of good writing, styles, barriers in written communication); internal communication (notices, memos, office circulars, office orders, etc.); report writings, using e-mails; preparation of other written documents (resume, letter of application, etc.); mixed communication (websites, Powerpoint presentations); performance reviews, etc. Fundamentals of business writing (selection of words, construction of paragraphs, etc.); writing for effect; the characteristics of a business letter (organisation, structure and layout); 7Cs of business writing (grammar in context, correct use of tense, adverbs and prepositions, phrasal verbs, affixes, synonyms, antonyms, homonyms, hyponyms, etc.); Ethics, Plagiarism and Reliable Sources.

UNIT IV **SOCIAL COMMUNICATION -** Social Communication – Etiquette in LSRW – polite yet assertive, tackling questions, seeking permission, expressing gratitude; body language – voice, posture and gesture; eye contact; selling, advertising and persuading – understanding cultural codes.; Non – Verbal communication – Appearance, Body language, Para language, Time, Space, Silence**.**

UNIT V **BUSINESS COMMUNICATION: CASE STUDIES -** Analysis of samples of authentic journal entries, memos, e-mails, web pages, audio recordings, taped transcripts, etc., and checking for clarity, conciseness, correctness, etc.

TEXT BOOKS:

1. Monippally, Matthukutty. M., "Business Communication Strategies", 11th Reprint, Tata McGraw-Hill. New Delhi, 2001
2. McLean, Scott., **"**Business Communication for Success.", Publisher: Flat World Knowledge, 2010 (Free e-book)

REFERENCE BOOKS:

1. Booher, Dianna, "E-Writing: 21st Century Tools for Effective Communication**"** New York: Pocket Books, a division of Simon & Schuster, Inc., 2001 (paper)
2. Guffey, Mary Ellen., "E-book--Business Communication: Process and Product" Fifth edition, Cincinnati, Ohio: South-Western College Publishing, An International Thomson Publishing Company, 2006

NOTE: The students will be given a reading list, including URLs to OERs

SEMESTER - III

DIF 7301 APPLICATION SECURITY Credits: 3

COURSE OBJECTIVES

- Become aware of the basics of cryptography and security
- Understand protocols and access issues
- Understand common vulnerabilities and their mitigation
- Understand some of the application aspects of security

COURSE OUTCOMES

- Able to understand common network vulnerabilities and provide for their mitigation
- Able to design, deploy, monitor, test and manage secure systems on the web

UNIT I **INTRODUCTION -** Security Engineering – Framework, Applications. Application Usability – Passwords (attacks and countermeasures), System Issues. Basic Cryptography -- History, Random Model, Symmetric Cryptography, Modes of Operation, Hash Functions, Asymmetric Cryptography-PKI.

UNIT II **PROTOCOLS AND ACCESS CONTROLS -** Eavesdropping, Simple Authentication, Authorization, Encryption Management, Formal Verification, OS access control, Hardware Protection, Access Control Failures.

UNIT III **COMMON APPLICATION VULNERABILITIES & MITIGATION IN WEB APPLICATION -** Introduction to the application vulnerability & Mitigations -- Injection , Broken Authentication and Session Management, Cross-Site Scripting (XSS), Insecure Direct Object References , Security Misconfiguration, Sensitive Data Exposure, Missing Function Level Access Control, Cross-Site Request Forgery (CSRF), Using Known Vulnerable Components, Unvalidated Redirects and Forwards, Debugging Tools – burpsuite, Tamper Data, Apache Jmeter.

UNIT IV **MULTILEVEL AND MULTILATERAL SECURITY -** Multilevel Security Policy Models, Real Life Examples – SELinux, AppArmor, Virtualization. Problems. Multilateral Security – Models – Lattice, Chinese Wall, BMA. Inference Control.

UNIT V **SECURE SYSTEMS DEVELOPMENT AND EVALUATION -** Security Management – Risk Management, Organisational issues, Methodologies, Security Risk Engineering. System Assurance. Evaluation – Common Criteria (ISO/IEC15408), hostile review, FOSS, CERTS, Bugtraq, Education, Security audit.

TEXT BOOKS:

1. Ross Anderson, “Security Engineering”, Wiley, 2nd Edition, 2008. http://www.cl.cam.ac.uk/~rja14/book.html, Digital copy available for free download
2. “The Open Web Application Security Project Development Guide”, https://www.owasp.org/index.php/Category:OWASP_Guide_Project (for Unit III)

REFERENCE BOOKS :

1. Alfred J. Menezes, Paul C. van Oorschot and Scott A. Vanstone, “Handbook of Applied Cryptography”, http://cacr.uwaterloo.ca/hac/, Digital copy available for download
2. Matt Bishop, “Computer Security: Art and Science”, Addison-Wesley Professional, 2003
3. “Vulnerability Scanner and Web application Penetration testing”, Metasploit (Community Version), http://www.metasploit.com
4. Tony Howlett, “Open Source Security Tools- Practical Applications for Security”, Prentice Hall, 2004

DIF 7302 WEB DEVELOPMENT (WITH LAB COMPONENT) Credits: 5

COURSE OBJECTIVES

- Understand issues on the client side
- Understand issues on the server side
- Learn some techniques from web 2.0
- Get a feel of the practical aspects through the lab practice

COURSE OUTCOMES

- Theoretical and Practical skills in designing and deploying websites and web-based applications using FOSS technologies and tools
- Strong abilities in Javascript

UNIT I **INTRODUCTION TO WEB DEVELOPMENT -** Many facets of web development -- web pages, web applications, social network services, UI design, content management, deployment, stress testing, security analysis, search engine optimization.

UNIT II **CLIENT SIDE DEVELOPMENT -** Browsers – Web and Mobile, HTML5, CSS3 with Responsive Web Design, JavaScript, JSON, Ajax, jQuery, jQuery UI, jQuery Mobile, sizzle, QUnit, Phone Gap (Apache Cordova), Browser compatibility, Tools – editors, debuggers -- DOM inspector, firebug.

UNIT III **SERVER SIDE DEVELOPMENT -** Overview of Application Frameworks with various technologies – Java: Java EE, JavaScript: Node.js, Perl: Catalyst, PHP: Zend, Python: Django, Ruby: Rails. Detailed study of JavaScript: Node.js.

UNIT IV **DATA STORAGE -** RDBMS – Postgres, MySQL, SQLite. Need for NoSQL. Different types – Document stores – Apache CouchDB, Key-value stores – Cassandra, Redis, MongoDB.

UNIT V **WEB 2.0 TECHNOLOGIES -** History, Concepts – Rich Internet Application, Web Oriented Architecture, Social Web. Technologies. Usage. Web Services – Introduction, RESTful Services.

LAB UNIT JavaScript programming exercises covering topics from Units II and III and integrating them with the NoSQL databases of Unit IV.

TEXT BOOKS:

1. Robert W. Sebesta, “Programming with World Wide Web”, Pearson Education, 2008
2. “Web Development – A text book”, Compiled by NRCFOSS/AU-KBC Centre, 2013

REFERENCE BOOKS:

1. Garann Means, “Node for Front-End Developers -- Writing Server-Side JavaScript Applications”, O'Reilly Media, January 2012
2. Eric Redmond and Jim R. Wilson, “Seven Databases in Seven Weeks: A Guide to Modern Databases and the NoSQL Movement”, Pragmatic Programmers, 2012
3. “W3Schools Online Web Tutorials”, http://www.w3schools.com/

DIF 7303 APPLICATION DEVELOPMENT USING PHP (WITH LAB COMPONENT) Credits: 5

COURSE OBJECTIVES

- Get to know basics and advanced topics in PHP
- Understand PHP Libraries and Content Management Systems
- Learn about PHP app development framework

COURSE OUTCOMES

- Should be a good PHP application developer well conversant with PHP Libraries, CMS and Frameworks

UNIT I **INTRODUCTION TO PHP -** Introduction, Basics, Data types, Operators, Flow control, Arrays, Array functions, Strings and Regular expressions, Generators.

UNIT II **ADVANCED PHP -** OOP in PHP -- Classes, Objects, Constructors and Destructors, Access Modifiers, Methods, Inheritance, Error and Exceptional Handling , File Handling, PEAR, Security.

UNIT III **PHP LIBRARIES -** Common libraries – cURL, GetText, GD, DB access – MySQL, PostgreSQL, SQLite, XML handling using PHP Data Objects (PDO).

UNIT IV **PHP CMS -** CMS Introduction. Advantages of a CMS, Different types of CMS, Examples , Drupal -- Installation – Content Management, Structure – Site Building – Modules – Theming.

UNIT V **PHP APPLICATION FRAMEWORKS -** Web Development Frameworks – Introduction – Yii – Model View Controller – Entry Script – Application – Controller – Model – View – Component – Module, PHP application development pitfalls.

LAB UNIT Exercise from all the five Units

TEXT BOOK:

1. Kevin Tatroe, Peter MacIntyre,Rasmus Lerdorf, “Programming PHP”, Creating Dynamic Web Pages, O'Reilly Media, 3rd Edition, 2013

REFERENCE BOOKS:

1. http://php.net
2. http://www.tutorialspoint.com/php/index.htm

DIF 7304 MOBILE APPLICATION DEVELOPMENT Credits : 5
(WITH LAB COMPONENT)

COURSE OBJECTIVES

- Learn the characteristics of mobile applications
- Understand the intricacies of UI required by mobile applications
- Study about the design aspects of mobile application
- Learn development and programming of mobile applications on the Android platform

COURSE OUTCOMES

Upon Completion of the course, the students would be able to:

- Design and implement the user interfaces of mobile applications
- Design mobile applications that are aware of the resource constraints of the devices
- Develop advanced mobile applications that accesses the databases and the web
- Become an experts on Google Android development

UNIT I **INTRODUCTION TO MOBILE DEVELOPMENT -** What is mobile computing?, History of mobile environments – early mobile phones to smartphones and tablets, Development for mobile environments, Differences from traditional application development, Trends in mobile development.

UNIT II **MOBILE DEVELOPMENT -** Introduction, Advantages, Limitations, Features useful for mobiles – Geolocation, offline web applications, offline web storage, animations, 2D/3D graphics, Audio/Video etc. , Frameworks -- HTML5 , Phone Gap (Apache Cordova) framework and jQuery Mobile framework.

UNIT III **INTRODUCTION TO ANDROID -** Android Overview -- Features, Architecture, Applications, Application frameworks, Libraries, Runtime, Kernel, Android Ecosystem – Application stores, publishing, Android Development Tools – Android SDK, Android emulator, Development on hardware devices.

UNIT IV **BASIC ANDROID DEVELOPMENT -** Writing Android Applications, Activity Lifecycle, Multi device support, Fragments, Data storage, Intents, Data sharing, Audio playback, Photo capture.

UNIT V **ADVANCED ANDROID DEVELOPMENT -** Animations. OpenGL ES, Wireless connections, Data syncing, Location aware applications, Best practices for development, Security, Distribution and Monetizing.

LAB UNIT Exercises using PhoneGap and the Android SDK using various features of Android

TEXT BOOK:

1. Ed Burnette, “Hello Android: Introducing Google's Mobile Development Platform”, The Pragmatic Programmers, 3rd edition, 2010

REFERENCE BOOKS:

1. http://developer.android.com
2. http://www.html5rocks.com/en/mobile
3. http://mobilehtml5.org/

DIF 7305 COLLABORATIVE DEVELOPMENT WITH VERSION CONTROL Credits: 3

COURSE OBJECTIVES

- Understand Version Control concepts
- Learn about centralized and distributed Version Control
- Understand Version Control hosting

COURSE OUTCOMES

- Expertise in the use of FOSS Version control technologies, tools and applications, in both the Centralised and Distributed versions, including Subversion hosting

UNIT I **INTRODUCTION TO VERSION CONTROL -** History – RCS, CVS, SVN, Git, Mercurials etc., Concepts – Versioning, Repository, Types of Repositories, Repository structure (Trunk, Branch, Tag), Create, Checkout, Commit, Update, Add, Edit, Delete, Rename, Move, Status, Diff, Revert, Log, Tag, Branch, Merge, Resolve, Lock, Command line vs. GUI mode of Operation.

UNIT II **CENTRALIZED VERSION CONTROL -** Subversion Basics – SVN AdminCreate, Checkout, Add, Status, Commit, Log, Diff, Update, Commit (with a merge) , Update (with merge) , Move , Rename, Delete, Lock, Revert, Tag, Branch, Merge (no conflicts), Merge (repeated, no conflicts), Merge (conflicts), Access Protocols (file://, http://, svn://, svn+ssh://).

UNIT III **DISTRIBUTED VERSION CONTROL BASICS -** DVCS Concepts – Distinguishing local and remote repository, Clone, Push, Pull, Directed Acyclic Graphs (DAGs), Advantages, Disadvantages. Workflows – managing multiple releases, shrinkwrap software, web software.

UNIT IV **DISTRIBUTED VERSION CONTROL -** Git Basics – Create, Clone, Add, Status, Commit, Push, Pull, Log, Diff, Update, Commit (with a merge), Update (with merge), Move, Rename, Delete, Revert, Tag, Branch, Merge (no conflicts), Merge (repeated, no conflicts), Merge (conflicts).

UNIT V **VERSION CONTROL HOSTING -** Setting up subversion hosting – svnserve, Apache (dav_module, viewvc). Git on the server – Configuration, gitweb, gitolite. Version control in hosted services – Sourceforge, Savannah, GitHub, Bitbucket, Gitorious.

TEXT BOOK:

1. Eric Sink, “Version Control by Example”, Pyrenean Gold Press, 2011. http://www.ericsink.com/vcbe/index.html Available in digital form for free download

REFERENCE BOOKS:

1. Version control with subversion – http://svnbook.red-bean.com
2. Pro Git – http://git-scm.com/book

SEMESTER - IV

ELECTIVES

DIF 7001 VIRTUALISATION AND CLOUD COMPUTING Credits : 3

COURSE OBJECTIVES

- Evolution of cloud computing
- Understand virtualization concepts
- Building and Programming clouds
- Become aware of the cloud security issues

COURSE OUTCOMES

- Good grasp of the theory and applications of virtualization
- Ability to build and program FOSS clouds using Eucalyptus etc
- Good understanding of the issue of security in the cloud environment and techniques of providing the same

UNIT I **INTRODUCTION -** Evolution of Cloud Computing –System Models for Distributed and Cloud Computing – NIST Cloud Computing Reference Architecture – Infrastructure as a Service (IaaS) – Resource Virtualization – Platform as a Service (PaaS) – Cloud platform & Management – Software as a Service (SaaS) – Available Service Providers.

UNIT II **VIRTUALIZATION -** Basics of virtualization - Types of Virtualization - Implementation Levels of Virtualization - Virtualization Structures - Tools and Mechanisms - Virtualization of CPU, Memory, I/O Devices - Desktop virtualization – Server Virtualization – Linux KVM, Xen, Qemu, LXC, OpenVZ.

UNIT III **CLOUD INFRASTRUCTURE -** FOSS Cloud Software Environments - Eucalyptus, Open nebula, OpenStack – OpenStack Architecture – Compute, Object Storage, Image Service, Identity, Dashboard, Networking, Block Storage, Metering, Basic Cloud Orchestration and Service Definition.

UNIT IV **CLOUD PROGRAMMING MODEL -** Parallel and Distributed programming Paradigms – MapReduce , Twister and Iterative MapReduce – Mapping Applications - Programming Support – Apache Hadoop – HDFS, Hadoop I/O, Hadoop configuration, MapReduce on Hadoop.

UNIT V **SECURITY IN THE CLOUD -** Security Overview – Cloud Security Challenges – Software-as-a-Service Security – Security Governance – Risk Management – Security Monitoring – Security Architecture Design – Data Security – Application Security – Virtual Machine Security – Qubes – Desktop security through Virtualization.

TEXT BOOKS:

1. Kai Hwang, Geoffrey C Fox, Jack G Dongarra, “Distributed and Cloud Computing From Parallel Processing to the Internet of Things”, Morgan Kaufmann Publishers, 2012
2. John W.Rittinghouse and James F.Ransome, “Cloud Computing: Implementation, Management, and Security”, by: CRC Press 2010

REFERENCE BOOKS:

1. Toby Velte, Anthony Velte, Robert Elsenpeter, “Cloud Computing, A Practical Approach”, TMH, 2013
2. George Reese, “Cloud Application Architectures: Building Applications and Infrastructure in the Cloud: Transactional Systems for EC2 and Beyond (Theory in Practice (O'Reilly)) “, O'Reilly
3. James E. Smith, Ravi Nair, “Virtual Machines: Versatile Platforms for Systems and Processes, Elsevier/Morgan Kaufmann” , 2005
4. Katarina Stanoevska-Slabeva, Thomas Wozniak, Santi Ristol, “Grid and Cloud Computing – A Business Perspective on Technology and Applications”, Springer http://docs.openstack.org/ops/ – Openstack Operations Guide
5. Tom White, “Hadoop: The Definitive Guide”, O'Reilly Media, 2009

DIF 7002 REALTIME AND EMBDEDDED PROGRAMMING Credits: 3

COURSE OBJECTIVES

- Understand the architecture of embedded processor, microcontroller and peripheral devices
- Able to program micro controllers in assembly for embedded systems
- Understand challenges of Real Time Operating Systems
- Programming embedded systems in high level languages such as C
- Get introduced to embedded Linux

COURSE OUTCOMES

Upon Completion of the course, the students would have the:

- Ability to understand architecture of embedded processors and microcontroller, and write Assembly and C language programs on them
- Ability to interface embedded systems and RT Linux to peripherals using typical interfacing standards
- Design and Development of Embedded system prototypes

UNIT I **INTRODUCTION -** Introduction – Uses – Limitations – Characteristics – User Interface – Microprocessors – Single Board Computers – System on a Chip – Peripherals – Tools – Reliability. Common SBCs – Beagle Board – Arduino – Raspberry Pi.

UNIT II **MICROPROCESSOR BASICS -** Microcontrollers – Microprocessors – Architecture – Von Neumann – Harvard – RISC – Programmable Controllers – Floating Point and Fixed Point Numbers – Memory – Memory Units – Microprocessor Families – Intel – ARM.

UNIT III **PROGRAMMING EMBEDDED SYSTEMS -** Common embedded software architectures – Simple Control Loop – Interrupt Controlled System – Cooperative Multitasking – Pre-emptive Multitasking – Mixed language programming – C and Assembly – I/O Programming – Serial and Parallel I/O – Boot loaders – TSR.

UNIT IV **REAL TIME OPERATING SYSTEMS -** Design Philosophies – Threading and Synchronization – Interrupts – Locks and Critical Sections – Implementation – CPU Scheduling – Memory Management – Common RTOS – Real Time Linux.

UNIT V **EMBEDDED LINUX -** Basic Concepts – Development Tools – Kernel Selection – Filesystem setup – Networking Services – Debugging Tools.

TEXT BOOKS:

1. http://en.wikibooks.org/wiki/Embedded_Systems – Embedded Systems Text book
2. Karim Yaghmour, Jon Masters, Gilad Ben-Yossef, Philippe Gerum, “Building Embedded Linux Systems, Concepts, techniques, tricks, and traps.”, 2nd Edition, O'Reilly Media., 2008

REFERENCE BOOKS:

1. http://nptel.iitm.ac.in/courses/Webcourse-contents/IIT%20Kharagpur/Embedded%20systems/New_index1.html – NPTEL course on Embedded Systems

DIF 7003 SOFTWARE TESTING Credits - 3

COURSE OBJECTIVES

- Introduce the basics and necessity of Software Testing and debugging
- Introduce various testing techniques along with the software production process
- Learn about common FOSS Testing Tools

COURSE OUTCOMES

Upon Completion of the course, the students would be able to:

- Perform automated testing using FOSS test tools
- Document and Manage the testing procedures

UNIT I **INTRODUCTION -** Software Testing background – software bugs- cost of bugs-software testing realities- Testing Axioms – Precision and Accuracy-verification and validation- quality and reliability-testing and quality assurance.

UNIT II **SOFTWARE TESTING METHODOLOGY -** Functional testing - Structural testing – Static and Dynamic testing – low level specification test techniques – Equivalence Partitioning – Data testing – State Testing – formal reviews – coding standards and guidelines – code review checklist – data coverage- code coverage – xUnit frameworks for various languages – Performance testing and Security Testing.

UNIT III **TEST MANAGEMENT -** Goal of Test Planning – test phases – test strategy – resource requirements – test schedule – writing and tracking test cases- Bug tracking systems – Bugzilla – Testopia– metrics and statistics- risks and issues – *T*est Data management – Traceability – Root Cause Analysis.

UNIT IV **AUTOMATED TESTING AND TEST TOOLS -** Benefits of automation and tools – TestLink – viewers and monitors – drivers – stubs – stress and load tools – Apache Jmeter – analysis tools- software test automation – random testing – beta testing.

UNIT V **SOFTWARE TESTING LIFECYCLE -** Functional Testing – Selenium – Selenium Webdriver – Watir – Sahi – Continuous Integration – Cruise Control – Hudson – Acceptance Testing – Concordion – Cucumber – Fitnesse – Project management and Tracking – Xplanner – Redmine – Jira – Defect management – Bugzilla – Testopia.

TEXT BOOK:

1. Glenford J.Myers, Tom Badgett, Corey Sandler, “The Art of Software Testing”,3rd edition, John Wiley & Sons publication, 2012
2. Ron Patton, “Software testing”, second edition, Pearson education, 2009
3. Srinivasan Desikan, Gopalaswamy Ramesh, “Software testing- Principles and Practices”, Pearson education, 2009
4. AU-KBC Centre, “Software Testing using FOSS”, Textbook compiled and edited from available material, 2013

REFERENCE BOOKS:

1. Boris Beizer, “Software testing techniques”, DreamTech Press, 2009
2. www.opensourcetesting.org – Open source software testing tools, news and discussion

DIF 7004 PYTHON PROGRAMMING Credits : 3

COURSE OBJECTIVES

- Get introduced to Python
- Understanding data structures and modules in python
- Objects in Python
- Learn the Python library

COURSE OUTCOMES

- Become a good Python programmer who can provide cost-effective solutions for web applications development using the Python language

UNIT I **INTRODUCTION -** Introduction to Python – Installation – Python Interpreter – usage and customization – Editor setup – Variables, Expressions and Statements – Conditionals – Functions.

UNIT II **DATA STRUCTURES -** Strings – Lists – List Comprehensions – Stacks – Queues – Tuples – Sequences – Sets – Dictionaries – Sets.

UNIT III **MODULES, I/O AND EXCEPTION HANDLING -** Modules – Search path – Compiled modules – Standard modules – Packages – Input and Output functions – Files – read and write – Exception – Handling and Raising – User defined Exceptions.

UNIT IV **OOPS IN PYTHON -** OOPS in Python – Classes – Scopes and Namespaces – Class Objects – Instance Objects – Method Objects – Inheritance – Iterators – Generators – Generator Expressions.

UNIT V **PYTHON STANDARD LIBRARY -** OS Interface – Command line arguments – String Pattern Matching – Mathematics – Internet Access – Dates and Times – Data Compression – Performance Measurement – Quality Control – Templating – Multi-threading – Logging.

TEXT BOOKS:

1. The Python Tutorial available at http://docs.python.org/3.3/tutorial/
5. Peter Wentworth Jeffrey Elkner, Allen B. Downey, and Chris Meyers, “How to Think Like a Computer Scientist: Learning with Python”, 3rd edition, Free Online Version: http://openbookproject.net/thinkcs/python/english3e/

REFERENCE BOOKS:

1. Python Documentation available at http://www.python.org/doc/
2. A Byte of Python by Swarooph CH available at http://swaroopch.com/notes/python/