<table>
<thead>
<tr>
<th>Code No.</th>
<th>Course Title</th>
<th>Credits</th>
</tr>
</thead>
<tbody>
<tr>
<td>DIF 7101</td>
<td>Introduction to Computing</td>
<td>3</td>
</tr>
<tr>
<td>DIF 7102</td>
<td>Philosophy and practice of FOSS</td>
<td>3</td>
</tr>
<tr>
<td>DIF 7103</td>
<td>Basics of GNU/Linux Operating System &amp; FOSS tools (with Lab component)</td>
<td>5</td>
</tr>
<tr>
<td>DIF 7104</td>
<td>Object Oriented Programming with Data Structures (with Lab component)</td>
<td>5</td>
</tr>
<tr>
<td>DIF 7105</td>
<td>HTML, CSS, and Javascript (with Lab component)</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td><strong>TOTAL</strong></td>
<td><strong>21</strong></td>
</tr>
</tbody>
</table>

**SEMESTER - II**

<table>
<thead>
<tr>
<th>Code No.</th>
<th>Course Title</th>
<th>Credits</th>
</tr>
</thead>
<tbody>
<tr>
<td>DIF 7201</td>
<td>Introduction to Java Technology (with Lab component)</td>
<td>5</td>
</tr>
<tr>
<td>DIF 7202</td>
<td>Software Development Practices (with Lab component)</td>
<td>5</td>
</tr>
<tr>
<td>DIF 7203</td>
<td>RDBMS Basics (with Lab component)</td>
<td>5</td>
</tr>
<tr>
<td>DIF 7204</td>
<td>Computer Networks and the Internet</td>
<td>3</td>
</tr>
<tr>
<td>DIF 7205</td>
<td>Business Communications in English</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td><strong>TOTAL</strong></td>
<td><strong>21</strong></td>
</tr>
</tbody>
</table>

**SEMESTER - III**

<table>
<thead>
<tr>
<th>Code No.</th>
<th>Course Title</th>
<th>Credits</th>
</tr>
</thead>
<tbody>
<tr>
<td>DIF 7301</td>
<td>Application Security</td>
<td>3</td>
</tr>
<tr>
<td>DIF 7302</td>
<td>Web Development (with Lab component)</td>
<td>5</td>
</tr>
<tr>
<td>DIF 7303</td>
<td>Application Development using PHP (with Lab component)</td>
<td>5</td>
</tr>
<tr>
<td>DIF 7304</td>
<td>Mobile Application Development (with Lab component)</td>
<td>5</td>
</tr>
<tr>
<td>DIF 7305</td>
<td>Collaborative development with Version Control</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td><strong>TOTAL</strong></td>
<td><strong>21</strong></td>
</tr>
</tbody>
</table>

**SEMESTER - IV**

<table>
<thead>
<tr>
<th>Code No.</th>
<th>Course Title</th>
<th>Credits</th>
</tr>
</thead>
<tbody>
<tr>
<td>E1</td>
<td>Elective – I</td>
<td>3</td>
</tr>
<tr>
<td>E2</td>
<td>Elective – II</td>
<td>3</td>
</tr>
<tr>
<td>DIF 7411</td>
<td>Project Work</td>
<td>12</td>
</tr>
<tr>
<td></td>
<td><strong>TOTAL</strong></td>
<td><strong>18</strong></td>
</tr>
</tbody>
</table>
## List of Electives

<table>
<thead>
<tr>
<th>Code No.</th>
<th>Course Title</th>
<th>Credits</th>
</tr>
</thead>
<tbody>
<tr>
<td>DIF 7001</td>
<td>Virtualisation and Cloud Computing</td>
<td>3</td>
</tr>
<tr>
<td>DIF 7002</td>
<td>Realtime and Embedded Programming</td>
<td>3</td>
</tr>
<tr>
<td>DIF 7003</td>
<td>Software Testing</td>
<td>3</td>
</tr>
<tr>
<td>DIF 7004</td>
<td>Python programming</td>
<td>3</td>
</tr>
</tbody>
</table>
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 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, sound cards, graphics cards.


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:


REFERENCE BOOKS:

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:


REFERENCE BOOKS:

1. [http://ftacademy.org/materials/fsm/1#1](http://ftacademy.org/materials/fsm/1#1)
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: configuring 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:


REFERENCE BOOKS:

OBJECT ORIENTED PROGRAMMING WITH DATA STRUCTURES
(WITH LAB COMPONENT)

DIF 7104 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 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.

TEXT BOOKS:


REFERENCE BOOKS:

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

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


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].

TEXT BOOKS:

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.


LAB UNIT Exercises from all the Units

TEXT BOOKS:

2. Java Tutorial online, http://docs.oracle.com/javase/tutorial/

REFERENCE BOOKS:

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


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


REFERENCE BOOKS:

COURSE OBJECTIVES

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

COURSE OUTCOMES

- Ability to build systems using FOSS RBMS systems like Postgres 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 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:


REFERENCE BOOKS:

1. Postgresql Manuals – http://www.postgresql.org/docs/
2. SQLite Documents – http://www.sqlite.org/docs.html
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.
2. Free/Open Source Software Network Infrastructure and Security, Gaurab Raj IOSN

REFERENCE BOOKS:

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:**


**REFERENCE BOOKS:**


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


TEXT BOOKS:


REFERENCE BOOKS:

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


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

REFERENCE BOOKS:
APPLICATION DEVELOPMENT USING PHP (WITH LAB COMPONENT)

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


UNIT V


LAB UNIT

Exercise from all the five Units

TEXT BOOK:


REFERENCE BOOKS:

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


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


REFERENCE BOOKS:

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 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).


TEXT BOOK:


REFERENCE BOOKS:

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 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 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.

TEXT BOOKS:


REFERENCE BOOKS:

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
   http://docs.openstack.org/ops/ – Openstack Operations Guide
DIF 7002 REALTIME AND EMBEDDED 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


TEXT BOOKS:


REFERENCE BOOKS:

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


TEXT BOOK:


REFERENCE BOOKS:

2. www.opensourcetesting.org – Open source software testing tools, news and discussion
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


TEXT BOOKS:

1. The Python Tutorial available at http://docs.python.org/3.3/tutorial/

REFERENCE BOOKS: