



**Rayat Shikshan Sanstha's
Yashwantrao Chavan Institute of Science,
(Autonomous)
Satara**

**SYLLABUS FOR
B. Voc. (Software Development)**

**Third Year SEMESTER V / VI
With effect from JUNE 2020
Year 2020-21 onwards**

SYLLABUS

VS 351 INTRODUCTION TO INFORMATION SECURITY

LEARNING OBJECTIVES:

- ✓ To aware of principles and protocols of internetworks
- ✓ To understand the basic issues in information security
- ✓ To understand the concept of ciphers and cryptography.
- ✓ To impart an idea on various ciphers

- ✓ To understand the concept of digital signatures and e-mail security policies
- ✓ To impart an idea on malicious softwares and remedies.

SYLLABUS

Module I: Information Security: Network security, Confidentiality, integrity, authentication, security policy, basic network security terminology, cryptography, symmetric encryption, substitution ciphers, transposition ciphers, steganography, Block ciphers, modes of operation, Data Encryption Standard, Public key cryptography, applications, strength and weakness, RSA algorithm, key distribution (concepts only).

Module II: Authentication, authentication methods, message digest, digital signatures, digital signature algorithm, DSS, E-mail security: Pretty Good Privacy, working of PGP, S/MIME, MIME, IP Security, Architecture, IPsec: strengths and benefits, IPv4, IPv6, ESP protocol, Web Security: Secure Socket layer, SSL session and connection.

Module III: Malicious Software, viruses, working of anti-virus software, worms, Trojans, spyware, firewall, characteristics of firewall, packet filters, application level gateways, firewall architecture, trusted systems.

Module IV: Security and Law:- Regulations in India. Information Technology Act 2000/2008. Cyber Crime and the IT Act 2000/2008. Indian Contract Act 1872, Indian Penal Code, Indian Copyright Act, Consumer Protection Act. Future Trends – The Law of Convergence.

LEARNING OUTCOMES:

On completion of this course student shall:

- ✓ Be aware of principles and protocols of internetworks
- ✓ Understand the basic issues in information security
- ✓ Understand the concept of ciphers and cryptography.
- ✓ To impart an idea on various ciphers

- ✓ Understand the concept of digital signatures and e-mail security policies
- ✓ Impart an idea on malicious softwares and remedies.

REFERENCES

- ❖ Brijendra Singh, *Cryptography & Network Security*, PHI.
- ❖ Pachghare, V.K., *Cryptography and Information Security*, PHI.

VS 352 PRINCIPLES OF MANAGEMENT

LEARNING OBJECTIVES:

To introduce students to:

- ✓ Concept of Management and Organisations
- ✓ Planning and decision making strategies
- ✓ Concepts of organizational behavior and HR management
- ✓ Leadership qualities

SYLLABUS

Module 1: Definition of Management – evolution of management principles - styles of Management – levels in management-structured and unstructured decision making – functions of management. Organizational behaviour – motivational theories

Module 2: Production & Marketing Management: Time management–workflow design – scheduling CP/M – critical path – PERT, Problems, Types of Markets – Marketing Mix – Product life cycle – pricing strategies – advertisement-sales promotion

Module 3: Quality Management. Concept of quality, total quality management, 7 sigma principles, ISO certifications, Component maturity models, CMM Levels.

Module 4: Human Resource Management: Meaning of HRM, Recruitment- selection and training – difference between training and development – on the job and off the job training.

REFERENCES

Core

- ❖ PC Tripathi and P N Reddy, *Principles of management*, 2/e, Tata McGraw Hill
- ❖ Poornima M. Charantimath, *Total Quality Management*, Pearson Education

Additional

- E H McGrath, *Basic Managerial Skills for All*, Prentice Hall of India

Internet resources:

- www.prenhall.com/robbins

LEARNING OUTCOMES:

On completion of this course student shall understand:

- ✓ Concept of Management and Organizations
- ✓ Planning and decision making strategies
- ✓ Concepts of organizational behavior and HR management
- ✓ Leadership qualities

LEARNING OBJECTIVES:

1. To familiarize the students with the latest programs of the government authorities in promoting small and medium industries.
2. To impart knowledge regarding how to start new ventures.

SYLLABUS

Module 1: Concepts of entrepreneur: Entrepreneur- Definitions-Characteristics of entrepreneur- Classification of entrepreneur-Entrepreneurial traits- Entrepreneurial functions- role of entrepreneurs in the economic development- Factor effecting entrepreneurial growth- Entrepreneurship - Meaning- definition- Entrepreneur Vs Intrapreneur- Women Entrepreneurs- Recent development-Problems-Entrepreneurial Development Programmes- Objectives of EDP- Methods of training- Phases of EDP

Module 2: Institutional support and incentives to entrepreneurs- Functions of Department of Industries and Commerce (DIC) - Activities of Small Industrial Development Corporation (SIDCO)-Functions of National Small Industries Corporation(NSIC)-Functions of Small Industries Development Bank of India (SIDBI)-Khadi Village Industry Commission (KVIC)- Small Industries Service Institute (SISI)- Functions and services of Kerala Industrial Technical Consultancy Organisation (KITCO)-Activities of Science and Technology Entrepreneurship Development Project (STEDP)-Strategies of National entrepreneurship Development Board(NEDB)-Objectives of National Institute for entrepreneurship and small business development (NIESBUD)- Techno park-Functions of techno park Incentives-Importance- Classification of incentives- Subsidy- Types of Subsidy

Module 3: Micro Small and Medium Enterprises- Features- Objectives- Importance- Role of SME in the economic development- MSME Act 2006- Salient features- Credit Guarantee Fund Trust Scheme for MSMEs - Industrial estates-Classification-Benefits-Green channel-Bridge capital- Seed capital assistance-Margin money schemes –Single Window System-Sickness- Causes –Remedies- Registration of SSI

Module 4: Setting up of Industrial unit-(Only Basic study) Environment for Entrepreneurship –Criteria for selecting particular project- Generating project ideas-Market and demand analysis-Feasibility study- Scope of technical feasibility- Financial feasibility- Social cost benefit analysis-Government regulations for project clearance- -Import of capital goods- approval of foreign collaboration-Pollution control clearances- Setting up of micro small and medium enterprises-Location decision- Significance.

Module 5: Project Report-Meaning-Definition-Purpose of project reports-Requirements of good report-Methods of reporting-General principles of a good reporting system-Performa of a project report-Sample project report.

REFERENCES

1. Shukla M.B. *Entrepreneurship and small Business Management*, KitabMahal Allahabad.
2. SangramKeshariMohanty, *Fundamentals of entrepreneurship*, PHI, New Delhi.
3. Nandan H. *Fundamentals of Entrepreneurship*, PHI, NewDelhi.
4. *Small-Scale Industries and Entrepreneurship*, Himalaya Publishing ,Delhi
5. C.N.Sontakki, *ProjectManagement*, Kalyani Publishers, Ludhiana.
6. SangamKeshariMohanty. *Fundamentals of Entrepreneurship*, PHI, NewDelhi
7. Peter F. Drucker- *Innovation and Entrepreneurship*.
8. *Vasanth Desai, Small Business Entrepreneurship, Himalaya Publications.*
9. *MSME Act 2006.*

LEARNING OUTCOMES:

On completion of this course student shall understand:

1. Familiarize the students with the latest programs of the government authorities in promoting small and medium industries.
2. Knowledge regarding how to start new ventures.

VS 354 PROGRAMMING IN JAVA

LEARNING OBJECTIVES:

- ✓ Let students install and work with JDK, also make them aware the use of java doc.
- ✓ Practice basic data types, operators and control structures in Java
- ✓ Practice basic handling of classes and objects in Java
- ✓ Introduce the following selected APIs: I/O, Strings, Threads, AWT, Applet, Networking
- ✓ Idea to approach and use a new package
- ✓

SYLLABUS

Module-I: Brief History of Java, Special Features of Java, Data Type & Operators in Java, Arrays, Objects, the Assignment Statement, Arithmetic Operators, Relational and Logical Operators in Java, control Structures, The Java Class, Constructor, Simple Java Application, simple Java Applet, Finalizers, Classes inside classes : composition.

Module-II: Inheritance & Interface, Deriving Classes, Method Over-riding, Method Overloading, Access Modifiers, Abstract Class and Method, Interfaces, Packages, Imports and Class Path.

Module-III: Exception Handling, The Try-Catch Statement, Catching more than one Exception, The Finally Clause, Generating Exceptions, Threads: Introduction, Creating Threads in Applications, Method in Thread Class, Threads in Applets.

Module-IV: Java APIs – overview of APIs, IO Packages, Java Input Stream Classes, Java Output Stream Classes, File Class, Graphic & Sound: AWT and Swing, Graphic methods, Fonts, Loading and Viewing Images, Loading and Playing Sound, AWT & Event Handling, Layouts, JDBC.

REFERENCES

Core

- ❖ *Java Programming, Schaum Outline Series*

Additional

- Deitel, *Java: How To Program*, Pearson Education

LEARNING OUTCOMES:

On completion of this course student shall understand:

- ✓ How to install and work with JDK, also make them aware the use of java doc.
- ✓ Practice basic data types, operators and control structures in Java
- ✓ Practice basic handling of classes and objects in Java
- ✓ APIs: I/O, Strings, Threads, AWT, Applet, Networking
- ✓ Idea to approach and use a new package

VS 355 DATABASE ADMINISTRATION

1. LEARNING OBJECTIVES

- ✓ To understand Database Environment.
- ✓ To understand commands, Database objects like – Views, indexes, sequences, synonyms and data dictionary.
- ✓ Object Oriented Database Management Systems Concepts.
- ✓ To understand RDBMS applications , Decision making processes

SYLLABUS

Module I: Introduction to Databases, Database Environment, The Relational Model and Languages: The Relational Model, Relational Algebra and Relational Calculus, QBE, Database Design:, ER diagrams and their transformation to relational design, Normalization, Security, Transaction Management: Transaction Support, Concurrency Control, Database Recovery.

Module II: Data Definition Language (DDL), Data Manipulation Language (DML), Data Control Language (DCL) commands, Database objects like – Views, indexes, sequences, synonyms and data dictionary, SQL.

Module III: Object Oriented Database Management Systems: Concepts, Composite objects, Integration with RDBMS applications, Issues in OODBMSs, Advantages and Disadvantages of OODBMSs

Module IV: Object-Oriented Database Design. Decision making processes, evaluation of DSS, Group decision support system. Distributed Databases

REFERENCES

- Database Administration: The Complete Guide to DBA Practices and Procedures by Craig S.Mullins
- MySQL Database Usage & Administration by VikramVaswani
- Understanding Database Administration by Pablo Berzakov

LEARNING OUTCOMES

At the end of this course, the students will be able to:

- ✓ Understand Database Environment.
- ✓ Understand commands, Database objects like – Views, indexes, sequences, synonyms and data dictionary.
- ✓ Object Oriented Database Management Systems Concepts.
- ✓ Understand RDBMS applications , Decision making processes

VS 356 Python

LEARNING OBJECTIVES:

- ✓ Learn to program and programming paradigms brought in by Python with a focus on File Handling and Regular Expressions How Appreciate the importance of having a process for software development.

SYLLABUS:

Module 1: Introduction : Introduction, What is Python, Origin, Comparison, Comments, Operators, Variables and Assignment, Numbers, Strings, Lists and Tuples, Dictionaries, if Statement, while Loop, for Loop and the range() Built-in Function, Files and the open() Built-in Function, Errors and Exceptions, Functions, Classes, Modules Syntax and Style Statements and Syntax, Variable Assignment, Identifiers, Basic Style Guidelines, Memory Management, Python Application Examples

Module II: Python Objects: Python Objects, Standard Types, Other Built-in Types, Internal Types, Standard Type Operators, Standard Type Built-in Functions, Categorizing the Standard Types, Unsupported Types. Numbers and Strings. Introduction to Numbers, Integers, Floating Point Real Numbers, Complex Numbers, Operators, Built-in Functions. Sequences: Strings, Lists, and Tuples, Sequences, Strings, Strings and Operators, String-only Operators, Built-in Functions, String Built-in Methods, Special Features of Strings

Module III: Lists & Dictionaries: Operators, Built-in Functions, List Type Built-in Methods, Special Features of Lists, Tuples, Tuple Operators and Built-in Functions, Special Features of Tuples. Introduction to Dictionaries, Operators, Built-in Functions, Built-in Methods, Dictionary Keys, if statement, else Statement, elif Statement, while Statement, for Statement, break Statement, continue Statement, pass Statement, else Statement

Module IV: Files and Input/Output: File Objects, File Built-in Function, File Built-in Methods, File Built-in Attributes, Standard Files, Command-line Arguments, File System, File Execution, Persistent Storage Modules. **Regular Expressions-** Introduction/Motivation, Special Symbols and Characters for REs, REs and Python **Errors and Exceptions-** What Are Exceptions? Exceptions in Python, Detecting and Handling Exceptions, Exceptions as Strings, Raising Exceptions, Assertions, Standard Exceptions Functions: Functions, Calling Functions, Creating Functions, Formal Arguments, Positional Arguments, Default Arguments Default Function Object Argument Example, Variable-length Arguments,

REFERENCES:

Core

Chun, J Wesley, Core Python Programming, Second Edition, Pearson, 2007 Reprint 2010

Additional

- Barry, Paul, Head First Python, 2nd Edition, O Rielly, 2010
- Lutz, Mark, Learning Python, 4th Edition, O Rielly, 2009

LEARNING OUTCOMES:

At the end of the course, the students should be able to :

- ✓ Able to walkthrough algorithm
- ✓ Improve programming skills
- ✓ Appreciate Python Programming Paradigm
- ✓ Hands on Regular Expression
- ✓ Ability to Text Processing scripts
- ✓ Write to file handling scripts

VS 357 JAVA LAB**LEARNING OBJECTIVES:**

This course will provide hands-on practice, under a variety of programming situations with a focus on writing, debugging and analyzing object oriented programs:

- ✓ basic data types and control structures in Java
- ✓ installing and using JDK
- ✓ writing applications and applets
- ✓ managing classes and objects in a variety of situations
- ✓ using i/o, string, threads and net APIs
- ✓ solving moderately complex problems involving the above.

SYLLABUS

The laboratory work will consist of 15-20 Experiments

Part A

1. Testing out and interpreting a variety of simple programs to demonstrate the syntax and use of the following features of the language: basic data types, operators and control structures.
2. Class definitions and usage involving variety of constructors and finalizers
3. Programs involving various kinds of inheritances,
4. Program involving Method Over-riding, Method Over-loading
5. Program involving Abstract Class and Methods

Part B

6. Program involving Interface,
7. Program to demonstrate creation and handling of packages, their imports and Class Path.
8. Programs involving a variety of Exception Handling situations
9. Program to define a class that generates Exceptions and using objects of the class.
10. Program involving creating and handling threads in applications and applets.
- 11-12: Programs to demonstrate methods of various i/o classes
1. Programs to demonstrate methods of string class

2. Program to demonstrate AWT/Swing graphic methods
3. Program for Loading and Viewing Images, Loading and Playing Sound
4. Programs to demonstrate various Layouts
- 17-18 Programs to demonstrate event handling
19. Program to demonstrate simple server-client (using a single m/c both as client and server)
20. Debugging programs involving syntactic and/or logical errors

LEARNING OUTCOMES:

At the end of the course, the students should be able to understand :

- ✓ basic data types and control structures in Java
- ✓ installing and using JDK
- ✓ writing applications and applets
- ✓ managing classes and objects in a variety of situations
- ✓ using i/o, string, threads and net APIs
- ✓ Solving moderately complex problems involving the above.

VS 358 STUDY TOUR

- Students should be taken for at least 3-5 day tour any of the software industry and the credits should be awarded based on the study tour report submitted

VS 359 Python Lab & MAJOR PROJECT (PHASE I)

Python Lab

1. Write a program that asks the user for his name and then welcomes him. The output should look like this : `Enter your name : Saksham Hello Saksham`

2. Write a program that prompts the user to input a Celsius temperature and outputs the equivalent temperature in Fahrenheit. The formula to convert the temperature is: $F = 9/5 C + 32$ where F is the Fahrenheit temperature and C is the Celsius temperature.

3. Write a program that prompts the user to input a Celsius temperature and outputs the equivalent temperature in Fahrenheit. The formula to convert the temperature is: $F = 9/5 C + 32$ where F is the Fahrenheit temperature and C is the Celsius temperature.

4. Write a program that accepts seconds from keyboard as integer. Your program should convert seconds in hours, minutes and seconds. Your output should like this :

```
Enter seconds : 13400
Hours: 3.0
Minutes: 43.0
Seconds: 20.0
```

5. Write a program that prompts the user to input three integers and outputs the largest.

6. Write a program that prompts the user to input number of calls and calculate the monthly telephone bills as per the following rule:

Minimum Rs. 200 for up to 100 calls.
 Plus Rs. 0.60 per call for next 50 calls.
 Plus Rs. 0.50 per call for next 50 calls.
 Plus Rs. 0.40 per call for any call beyond 200 calls.

7. The marks obtained by a student in 3 different subjects are input by the user. Your program should calculate the average of subjects and display the grade. The student gets a grade as per the following rules:

Average Grade	
90-100	A
80-89	B
70-79	C
60-69	D
0-59	F

8. Write a program that prompts the user to input a character and determine the character is vowel or consonent.

9. Write a program that prompts the user to input a number and prints its mulitiplication table

10. Write a program that prompts the user to input a number and reverse its digits. For example, the reverse of 12345 is 54321;

11. Write a program that asks the user to input a positive integer. Your program should find and display the sum of digits of number. For example, sum of digits of number 32518 is $3+2+5+1+8 = 19$.

12. Write a program that accepts a list from user and print the alternate element of list.

13. Write a program that input a string and ask user to delete a given word from a string.

14. Write a program that accepts a string from user. Your program should count and display number of vowels in that string.

15. Write a program that reads a string from keyboard and prints the unique words. Your program should convert input string to lower case.

MAJOR PROJECT (PHASE I)

- The guide lines given for the course VS368 has to be followed for this course.
- An analysis and design report should be submitted at the end of this course. This report should be valued internally and marks should be awarded.
- Up to coding should be finished in 5th semester. Coding should begin in the 6th semester

VS 361 HUMAN RESOURCES DEVELOPMENT

LEARNING OBJECTIVES:

- ✓ Importance--scope and objectives of HRM
- ✓ Knowing oneself- confidence building- defining strengths- thinking creatively- personal values-time and stress management
- ✓ Group discussion- mock Group Discussion using video recording- public speaking

SYLLABUS

Module I: Introduction to Human Resource Management—Importance--scope and objectives of HRM. Evolution of the concept of HRM- Approaches to HRM- Human Resource Development- HRD and competitive advantage- .

Module II: Personal Skills- Knowing oneself- confidence building- defining strengths- thinking creatively- personal values-time and stress management. Social Skills- Appropriate and contextual use of language- non-verbal communication- interpersonal skills- problem solving.

Module III: Personality Development-Personal grooming and business etiquettes, corporate etiquette, social etiquette and telephone etiquette, role play and body language.

Module IV: Presentation skills - Group discussion- mock Group Discussion using video recording- public speaking Professional skills - Organizational skills- team work- business and technical correspondence- job oriented skills-professional etiquettes

LEARNING OUTCOMES:

At the end of the course, the students should be able to understand:

- ✓ Importance--scope and objectives of HRM
- ✓ Knowing oneself- confidence building- defining strengths- thinking creatively- personal values-time and stress management
- ✓ Group discussion- mock Group Discussion using video recording- public speaking

REFERENCE

1. MatilaTreece: Successful communication: Allyun and Bacon Pubharkat.
2. Human Resource Management- Text and Cases-- VSP Rao
2. Jon Lisa Interatid skills in Tourist Travel Industry Longman Group Ltd.
3. Robert T. Reilly – Effective communication in tourist travel Industry Dilnas Publication.
4. Boves. Thill Business Communication Today Mcycans Hills Publication.
5. Dark Studying International Communication Sage Publication.
6. Murphy Hilderand Thomas Effective Business Communication McGraw Hill.

VS 362 FREE AND OPEN SOURCE SOFTWARES (FOSS)

LEARNING OBJECTIVES:

- ✓ To Explain the features of free & open source software
- ✓ To Familiarization with LINUX
- ✓ To Work Shell Programming
- ✓ To Demonstrate the working of Basic System Administration, File Systems, Linux Files

SYLLABUS

Module 1: Linux Basics: History of Linux: (History, FOSS, current Linux Distributions-Distros examples), Linux Operating System Layers, The Linux Shell (different kinds of shell), Process: (parent and child processes), Files and Directories (File Structure and directory structure), Interaction with System.

Module 2: Shells and Utilities: Getting Started With Shell Programming: The bash shell, Shell commands, The role of shells in the Linux environment, Other standard shells, Write a simple shell script - "Hello World!" (Starting a script with Shebang line (!#)),Comments in a script ,Setting up permissions on a script, Execute a script, Debug a script), Variables in shell : (Assign values to shell variables, Default shell variables value , Rules for Naming variable name, Display the value of shell variables, Quoting, The export statement, Unset shell and environment variables, Getting User Input Via Keyboard), Bash variable existence check, Customize the bash shell environments: Recalling command history, Path name expansion, Create and use aliases, The tilde expansion, Startup scripts-(Using aliases, Changing bash prompt, Setting shell options , Setting system wide shell options), Commonly Used Commands and Utilities (ls,rm,cat etc – at least 25 commands/utilities for example)

Module 3: Basic Administration: Basic System Administration (Run levels, User accounts), Kernel Administration: (Linux kernel sources, rebuilding kernel, installing kernel), Managing Users, Managing File Systems, Linux File Permissions, Devices and Modules (device drivers)

Module 4: Files Systems and Editing Files: File Systems, Linux Files, Directories and Archives, The vi editor, awk, sed. Software Management, Office and Database Applications, Graphics Tools and Multimedia, Mail and News Clients, Web, FTP, and Java Clients, Security: Encryption, Integrity Checks, and Signatures, Security-Enhanced Linux, Secure Shell and Kerberos, Firewalls

REFERENCES

- ✓ The Linux Command Line, 2nd Edition: A Complete Introduction Book by William E. Shotts Jr.
- ✓ Linux bible Book by Christopher Negus

LEARNING OOUTCOMES:

At the end of this course, the students will be able to:

- ✓ To Explain the features of free & open source software
- ✓ To Familiarization with LINUX
- ✓ To Work Shell Programming
- ✓ To Demonstrate the working of Basic System Administration, File Systems, Linux Files

VS 363 IT & SOCIETY**LEARNING OBJECTIVES:**

- ✓ To having a clear view of what professionalism is
- ✓ To increase awareness of ethical issues in computing profession
- ✓ To increase awareness of managing quality
- ✓ To increase awareness of quality certifications
- ✓ To have an Exposure to Cyber law

SYLLABUS

Module-I: What is a profession – who is a professional – core qualities of a professional – Environments and their impact and complexity – social attitudes, beliefs and values

Module –II: Codes of ethics - solving ethical conflicts, moral reasoning and ethical theories– responsibilities and rights. Computer ethics : ethics and the internet – hacking – netiquette – privacy

Module-III: Quality Management. Concept of quality, total quality management, 7 sigma principles, ISO certifications, Component maturity models, CMM Levels.

Module-IV Cyberlaw: Intellectual property rights – basic ideas – copyright concepts – copyrights applied to softwares – software licensing – patents in software – Indian copyright law and provisions for software – Indian patent law and provisions for software – various licencing models - arguments against copyrights and patents in software – free softwares – GPL software freedoms– open source softwares

REFERENCES**Core References**

- M Govindarajan, S Natarajan, V S Senthil Kumar, Engineering Ethics, PHI
- Poornima M. Charantimath, Total Quality Management, Pearson Education
- Richard Stallman, Free Software: A Perspective, Prajasakthi Book House, Hyderabad
- Indian Copyright Act and Indian Patent Acts

LEARNING OOUTCOMES:

At the end of this course, the students will be able to:

- ✓ Having a clear view of what professionalism is
- ✓ Aware of ethical issues in computing profession
- ✓ Aware of managing quality
- ✓ Aware of quality certifications
- ✓ Having an Exposure to Cyber law

VS 364 MOBILE APPLICATIONS DEVELOPMENT

LEARNING OBJECTIVES

- ✓ This course focuses on developing applications for modern mobile phone operating systems.
- ✓ Most of the course is dedicated to android and Microsoft's Windows.

SYLLABUS

Module I: Mobile Computing: Mobile Connectivity-Cells, Framework, wireless Delivery Technology and switching methods, Mobile Information Access Devices, Mobile Data Internetworking Standards, Cellular data communication Protocols, Mobile Computing Applications. Mobile Data Bases- Protocols, Scope, Tools and Technology, M- Business.

Module II: Android Development Tools: Eclipse integrated development environment (IDE), Android software development kit (SDK), Application development tools (ADT) plugin, Emulators and Devices: Android virtual devices (AVDs), Connecting Androids to the development platform, USB drivers for Android development, Android Development Lifecycle: "Hello World!", Running on the emulator, Running on a device

Module III: XML Fundamentals: Trees, Elements, Attributes, Examples, Simple Interactive programs, Activity Lifecycles: Callbacks and activity pyramids, Launcher activity, Instantiation, Destroying activities, Pausing, resuming, starting and stopping activities, Saving and restoring activities,

Module IV: Interaction with Other Apps, Location-Aware Apps, Network and Cloud Operations, Layout Hierarchies, Audio, Photos and Video

REFERENCES

- “Mobile Application Development: JavaScript Frameworks” by Hazem Saleh and Ethan Holmes Poornima M. Charantimath, Total Quality Management, Pearson Education
- Android App Development by Hervé J. Franceschi
- Cross-Platform Mobile Application Development: A Beginner's Guide Using Solar2D by John R. Carlson Ph.D.

LEARNING OUTCOMES:

At the end of this course, the students will be able to:

- ✓ How develop applications for modern mobile phone operating systems.
- ✓ How to solve problems regarding applications

VS 365 SOFTWARE TESTING

LEARNING OBJECTIVES:

- ✓ To Discuss the basic concept of testing
- ✓ To Explain different types of testing
- ✓ Describe the tools used for testing

SYLLABUS

Module-I: Introduction:- Purpose of testing, testing and debugging, model for testing, types of testing, types of bugs Flow graphs and Path testing:- Basics concepts of path testing, predicates, path predicates and achievable paths, path sensitizing, path instrumentation, **Module-II:** Transaction Flow Testing:-transaction flows, example of transaction testing. Dataflow testing:- Basics of dataflow testing, example of dataflow testing, Domain Testing:-domains and paths, Nice & ugly domains, Interfaces testing

Module-III: Paths, Path products and Regular expressions:- path products & path expression, reduction procedure, regular expressions & flow anomaly detection

Module-IV: Logic Based Testing:- overview, decision tables, path expressions, kv charts, State Graphs and Transition testing:- good & bad state graphs, state testing,

REFERENCES:

Core

- ❖ BarisBeizer, *Software Testing techniques*, Dreamtech, second edition.

Additional

- Edward Kit, *Software Testing in the Real World*, Pearson Education of India.
- K.K.Prasad, *Software Testing Tools*, Dreamtech

LEARNING OUTCOMES:

At the end of this course, the students will be able to

- ✓ Discuss the basic concept of testing
- ✓ Explain different types of testing
- ✓ Describe the tools used for testing

VS 366 ADVANCED JAVA

1. LEARNING OBJECTIVES

- ✓ To understand working of Java.
- ✓ To understand Java Servlets Development Kit, creating, Compiling and running servlet
- ✓ To understand Java Beans. Advantage, Installing / Starting/ Using BDK
- ✓ To understand JSP Architecture, JSP Access Model, JSP Syntax Basic

SYLLABUS

Module I: Introduction to Swing J applet, Icons and Labels, Text Fields, Buttons, JButton Class, Check Box, Radio Buttons, The Container, Panel, Windows and Frame Classes, Combo Box, Tabbed Panes, Scroll Panes, Trees, Tables, Custom Rendering of JList Cells, JDBC: JDBC Fundamentals, Establishing Connectivity and working with connection interface, Working with statements, Creating and Executing SQL statements, Working with Result Set Object & Result Set Meta Data

Module II: Servlets: Introduction to Servlets (Life cycle of servlets, Java Servlets Development Kit, creating, Compiling and running servlet, The servlet API : javax.servlet package, Reading the servlet Parameters, Reading Initialization parameter, The javax.servlet.http.Package, Handling HTTP Request and Response (GET / POST Request) vi) Using Cookies, Session Tracking

Module III: Introduction To Java Beans, What is Java Bean?, Advantage, Installing / Starting / Using BDK (Bean Development Kit, JAR Files, Using the Java Beans API

Module IV: JSP: Advantage of JSP technology (Comparison with ASP / Servlet), JSP Architecture, JSP Access Model, JSP Syntax Basic (Directions, Declarations, Expression, Scriptlets, Comments, JSP Implicit Object, Object Scope, Synchronization Issue, Exception Handling, Session Management

REFERENCES

- ✓ Advanced Java Programming Textbook by Uttam K. Roy.
- ✓ Java: Advanced Features and Programming Techniques Book by Clark Nathan

LEARNING OUTCOMES

At the end of this course, the students will be able to:

- ✓ Understand working of Java.
- ✓ Understand Java Servlets Development Kit, creating, Compiling and running servlet
- ✓ Understand Java Beans. Advantage, Installing / Starting/ Using BDK
- ✓ Understand JSP Architecture, JSP Access Model, JSP Syntax Basic

VS 367 ANDROID & JAVA Lab

- ❖ Familiarization with the Android Development Platform
 - Configure the settings of the Eclipse Integrated Development Environment (IDE) for Android development.
 - Use the IDE to create an Android programming project.
 - Write a short introductory program.
 - Connect to an Android device and run the program on the device.
- ❖ Writing Android Programs
 - Use the IDE to create an app project that has buttons, string resources, View and ViewGroup objects.
 - Experiment with XML, activities, manifests and intents in this project.
 - Use the emulator to test the app
 - Load the app onto an Android device and test it on actual hardware.
- ❖ Expanding the App Capabilities with Lifecycles and more UI options
 - Use the IDE to create an app that has an activity lifecycle including some of the following: pause, resume, start, stop, destroy and restore.
 - Experiment with UI fragments, and flexible UIs in this project.
 - Use the emulator to test the app
 - Load the app onto an Android device and test it on actual hardware.
- ❖ Demonstrating Inter-App Interaction
 - Use the IDE to create an app that has inter-application interaction.
 - Experiment with sending and getting messages to other apps in this project.
 - Use the emulator to test the app
 - Load the app onto an Android device and test it on actual hardware.
- ❖ Demonstrating Location Awareness
 - Use the IDE to create an app that is location-aware.
 - Experiment with the location manager in this project.
 - Use the emulator to test the app
 - Load the app onto an Android device and test it on actual hardware.
- ❖ Building a program that demonstrates layout hierarchies
 - Use the IDE to create an app that has multiple layout hierarchies.
 - Experiment with ListView, Lint, and ViewStubs in this project.

- Use the emulator to test the app
- Load the app onto an Android device and test it on actual hardware.
- ❖ Building a program that demonstrates network operations
 - Use the IDE to create an app that has network awareness.
 - Experiment network and cloud messaging in this project.
 - Use the emulator to test the app
 - Load the app onto an Android device and test it on actual hardware
- ❖ Incorporating audio and/or video into app projects
 - Use the IDE to create an app that has either audio or video (or both) incorporated into its design.
 - Experiment with volume, playback, photo-capture and/or video control in this project.
 - Use the emulator to test the app
 - Load the app onto an Android device and test it on actual hardware.
- ❖ Building a program around 2-D or 3-D graphics
 - Use the IDE to create an app that uses OpenGL ES with 2-D and/or 3-D graphics.
 - Experiment with projections, shapes, motion transformations and renderers in this project.
 - Use the emulator to test the app
 - Load the app onto an Android device and test it on actual hardware.

VS 368 MAJOR PROJECT & VIVA

LEARNING OBJECTIVES:

- ✓ To provide an opportunity to apply the knowledge gained through various courses in solving a real life problem
- ✓ To provide an opportunity to practice different phases of software/system development life cycle
- ✓ To introduce the student to a professional environment and/or style typical of a global IT industry
- ✓ To provide an opportunity for structured team work and project management

- ✓ To provide an opportunity for effective, real-life, technical documentation
- ✓ To provide an opportunity to practice time, resource and person management.

1. PROJECT GUIDELINES

- Group Size – Maximum 3
- No. of records – No. of group members + 1 (Department copy)
- Certificate should include the names of all members

The minimal phases for the project are: Project search, finalization and allocation, Investigation of system requirements, Data and Process Modelling, System Design, Program design, Program coding and unit testing, System integration, System implementation and acceptance testing.

3.1 Planning the Project: The Major Project is an involved Exercise which has to be planned well in advance. The topic should be chosen in Semester 4 itself and the case study of Course CS1302 should as far as possible, be based on the project topic, though on Exceptional cases, for valid reasons, the project guide may waive this condition. Related reading, training and discussions should start from semester 5 itself.

3.2 Selection of project work: Project work could be of 3 types:

a) Developing solution for a real-life problem: In this case, a requirement for developing a computer based solution already Exists and the different stages of system development life cycle is to be implemented successfully. Examples are Accounting Software Package for a particular organization, Computerisation of administrative functions of an organization, Web Based Commerce, etc. The scope for creativity and Exploration in such projects is limited, but if done meticulously, valuable Experience in the industrial context can be gained.

(b) Innovative Product development: These are projects where a clear-cut requirement for developing a computer based solution may not be Existing, but a possible utility for the same is conceived by the proposer. An Example is a Malayalam Language Editor with Spell Checker, Computer Music Software for Indian Music, Heat Engines Simulation Software for eLearning, Digital Water Marking Software,

(c) Research level project: These are projects which involve research and development and may not be as structured and clear cut as in the above case. Examples are Malayalam Character Recognition, Neural Net Based Speech Recogniser, Biometric Systems, Machine Translation System etc. These projects provide more challenging opportunities to students,

but at EX level is a difficult choice. If any student identifies proper support in terms of guidance, technology and references from External organizations and also the supervisors are convinced of the ability of the student(s) to take up the project, it shall be permitted. The methodology and reporting of such projects could be markedly different from type (a) and is left to the proposer/external supervisor of the projects.

3.3 Selection of Team: To meet the stated objectives, it is imperative that Major Project is done through a team effort. Though it would be ideal to select the team members at random (drawing lots) and this should be strongly recommended, due to practical considerations, students may also be given the choice of forming themselves into teams with 3 to 5 members (teams less than 3 members may be permitted in Exceptional cases, for valid reasons). A gender mix should also be strongly suggested. A team leader shall be elected through drawing lots. Teams shall maintain team meeting minutes and ensure that every team member has tasks assigned in writing. Team meeting minutes shall form a part of the Project Report. Even if students are doing projects as groups, each one must independently take up different modules of the work and must submit the reports also independently (though, in such cases, some common materials is permissible). Evaluation will also be done independently.

3.4 Selection of Tools: No restrictions shall be placed on the students in the choice of platforms/tools/languages to be utilized for their project work, though open source is strongly recommended, wherever possible. No value shall be placed on the use of tools in the evaluation of the project.

3.5 Selection of Organisation & Guide: No restrictions shall be placed on the students in the choice of organization where project work may be done, in terms of locality, type (public/private) etc. It is the duty of the Head of Institute/Principal of College to ensure that the Aim, Objectives and full project guidelines are communicated to the external organization. The guide should ideally be a post-graduate with minimum 2 years of work experience.

Students may also choose to do project in the college/institute (or partially in the college/institute and partially in an external organization), especially product-based work, but in such cases the supervisors must ensure that (i) industry practices are followed (ii) the

students undertake a planned visit to an IT industry with international operations to make up for the loss of experience and (iii) the services of an external guide with industry experience is obtained.

3.6 Project Management: Head of Institute/Principal of College should publish a list of students, projects topics, internal guide and external organization (if any) and teams agreed, before the end of semester 5. Changes in this list may be permitted for valid reasons and shall be considered favourably by Head of Institute/Principal of College any time before commencement of the project. Any request for change after commencement should be considered by a committee of 3 teachers and their recommendation shall be accepted by Head of Institute/Principal of College.

Gantt-chart of proposed activities and a draft statement of project deliverables (which may subsequently be altered if justified) should be prepared before the commencement of the project. The actual completion of each phase should be noted on the chart in the course of the project work. Students should submit a fortnightly report of progress which could be indication of percentage of completion marked on the original Gantt-chart, with any notes attached. Students should ideally keep a daily activity log sheet. Team meetings should be documented in the format given at the end. Changes in the submitted documents are possible, as project development is essentially an evolutionary process. The project guide must ensure that changes are necessary due to the knowledge gained in succeeding phases of the project. The date of completion of a phase should be brought forward if the changes made are deemed to be errors and not due to additional knowledge gained from a succeeding phase.

3.7 Documentation:

The following are the major guidelines: The final outer dimensions of the report shall be 21 cm X 30 cm. The colour of the flap cover shall be light green. Only hard binding should be done, with title of the thesis and the words —<BRIEF TITLE> BSc (CS) Project Report 200... displayed on the spine in 20 point, Bold, Times New Roman, as in example below. In case the title is too long, a shorter version of it may be used (Like —Image Pro instead of Image Pro – An Interactive Image Processing package). It is highly recommended that LaTeX be used for documentation.

- The text of the report should be set in 12 pt, Times New Roman, Single Spaced.

- Headings should be set as follows: CHAPTER HEADINGS 20 pt, Times New Roman, Bold, All Caps, Centered.

WEB BASED BILLING SOFTWARE: B.Voc.(S/D) PROJECT 2009

1. SECTION HEADINGS 12 pt, Times New Roman, Bold, All Caps, Left Adjusted.

1.1 Section Sub-headings 12 pt, Times New Roman, Bold, Left Adjusted.

Titles of Figures, Tables etc are done in 12 point, times New Roman, Italics, Centered.

<PROJECT TITLE>

<STUDENT'S NAME>

<COLLEGE NAME>

PROJECT REPORT

Submitted in partial fulfilment of the

Requirements for the award of

B.Voc (Software Development)

degree of Shivaji University of

Kourlha

2020

Some general guidelines on documentation stylistics are:

- Double quotes and single quotes (—|, —) should be used only when essential. In most cases words put in quotes are better highlighted by setting them in italics. Eg: This process is known as —morphing|. This process is known as *morphing*.
- Page numbers shall be set at right hand top corner, paragraph indent shall be set as 3.
- Only single space need be left above a section or sub-section heading and no space may be left after them.
- Certificate should be in the format: —Certified that this report titled..... is a bonafide record of the project work done by Sri/Kumunder our supervision and guidance, towards partial fulfillment of the requirements for the award of the Degree of BSC (Computer Science) of the University of Keralal with dated signatures of Internal; Guide, external guide and also Head of Institute/College.
- If the project is done in an external organization, another certificates on the letterhead of the organization is required: —Certified that his report titled. is a bonafide

record of the project work done by Sri/Kum..... under any supervision and guidance, at theDepartment of..... (Organization) towards partial fulfilment of the requirements for the award of the Degree of BSC (Computer Science) of the University of Kerala.

- References shall be IEEE format (see any IEEE magazine or transaction). Take care in use of italics and punctuation. While doing the project, keep note of all books you refer, in the correct format, and include them in alphabetical order in your reference list. Eg: A book is cited as: Kartalopoulos, S V Understanding Neural Networks and Fuzzy Logic, BPB Publishers, 1996, pp. 21-27. (pp.21-27 indicates that pages 21-27 have been referred. If the whole book is being referred, this may be omitted. If a single page is referred, say 7, it may be cited as p.7 Report writing is NOT a hasty activity done after finishing the project. Students must try to develop the report along with the work, so as to give it flesh and blood. Drafts should be read, modified, spell checked and grammar checked at least thrice during the course of the project and before a final printout is taken, the same may be got approved from the internal guide. The students should send two interim reports to internal guides. This will also help the students in their report writing.

The Gantt chart, fortnightly progress reports, and team meeting minutes mentioned in section 3.5 should appear as appendix to the project report. Regarding the body of the report, as an indicative Example, the following is given (though students should not attempt to fit every kind of project report into this format):

- Organizational overview (of the client organization, where applicable)
- Description of the present system
- Limitations of the present system
- The Proposed system- Its advantages and features
- Context diagram of the proposed system.
- Top level DFD of the proposed system with at least one additional level of Expansion
- Structure Chart of the System
- System flowchart
- Menu Tree
- Program List
- Files or tables (for DBMS projects) list. Class names to be entered for each file in OO systems.

- List of fields or attributes (for DBMS projects) in each file or table.
- Program – File table that shows the files/tables used by each program and the files are read, written to, updated, queried or reports were produced from them.
- Reports List with column headings and summary information for each report.
- System Coding and variable/file/table naming conventions
- System controls and standards
- Screen layouts for each data entry screen.
- Report formats for each report.

Program documentation is suggested on the following lines:

- Program id
- Program level run chart
- Program function Explanation
- Data entry screen (reproduced from system documentation).
- Report layout (reproduced from system documentations)
- Program level pseudo code or flowchart.
- Decision tables, decision trees, with English Explanation where necessary.
- Program listing
- Test data
- Test results.

3.8 Methodology:

Wherever applicable, object oriented approach should be used for software development. The project report should generally contain details of the following steps (*though students should not attempt to fit every kind of project into this format*):

(a) Analysis

- Study of existing systems and its drawbacks (general)
- Understanding the functionalities of the system (detailed)
- Preparation of requirement
- Conduct of Feasibility study
- Identification of relevant Objects
- Abstraction of each object (attributed and methods)
- Relationship between objects

(b) Design

- Design of each subsystems

- Design of each classes
- Design of communications between objects
- Design of Algorithms for problem solving
- User interface Design
- Any other steps if necessary

(c) Coding and Impletion

(d) Testing

(e) Security, Backup and Recovery Mechanisms

(f) On line help and User Manuals

(g) Upgradability Possibilities

3.9 Project IPR & Utilisation: The intellectual property rights in all project work done by the students shall vest with the University of Kerala, except in cases where some external organizations seek undertaking from students to concede IPR in all work done in their organization or under their guidance. Where possible, students should attempt to obtain at least a joint IPR for the University. In cases where project works are of public utility, students shall be asked to publish their work including source code and documentation, in so far as their rights are clear.

Evaluation of Project

- Documentation – 80 marks
- Content & Methodology - 80 marks
- Viva Voce - 80 marks

4. REFERENCES

4.1 Core

- ❖ S A Kelkar, *Software Project Management*, Prentice Hall of India
- ❖ W Alan Randolph, Barry Z. Posner, *Effective project planning and management*, PHI

4.2 Additional

- Greg Mandanis, *Software Project Management Kit for Dummies*, IDG Books
- Joel Henry, *Software Project management*
- Frederic P B, *Mythical Man-month, Essays on Software Engineering*, Addison Wesley
- David Lamport, *Latex: A document Preparation System, 2/e*, Pearson Education