University of Mumbai

T.Y B. Sc.
(Computer Science)

Syllabus

(W.E.F. 2005-2006)

(T.Y. B. Sc. Computer Science Syllabus)

There are Six Units in each paper. If there are two sections, each section of the paper is divided in 3 Units. The detailed references are mentioned at the end of each section instead of merely giving a list of reference book at the end of the paper.

It is expected that the question paper will have twelve questions, two in each unit, with internal option. The student will be required to attempt six questions, one in each unit. Thus the study of any unit in the paper may not be left as an option.

1. Syllabus details gives a list of practicals, with minimum number of practicals to be performed.

2. For the purpose of examination, the breakup of total marks for practicals (200 for Group I to IV and 80 for Applied Component) will be as follows:

A) Main subject Practical Examination:

Experiment I (Group I Practical)	: 45 Marks
Experiment II (Group II Practical)	: 45 Marks
Experiment III (Group III Practical)	: 45 Marks.
Experiment IV (Group IV Practical -PROJECT)	: 50 Marks.
Journal(Group I to III) & Viva-voce on Journal	: 15 Marks.
Total	200 Marks.

B) Applied Component Practical Examination:

Experiment I	: 35 Marks.
Experiment II	: 35* Marks.
Journal & Viva-voce on Journal	: 10 Marks.
Total	80 Marks.

Note: (a) Expt. II examination will be in the form of demonstration and Viva-voce on project, as mentioned in the syllabus.

(b) Each student must maintain a record of the experiments & projects performed, as per the syllabus and must bring the certified journals and project reports, duly signed by the teacher concerned and the Head of the Dept. at the time of final examination.

3. Requirement of the batch size, number of students per computer in a batch and the network configuration and other details are same as mentioned in Annexure III of the earlier report of the committee.

4. As regards the lab requirement (Hardware & Software), space and time table requirements are satisfied; the colleges may have multiple batches at T.Y.B.Sc. (Computer Science) Class.

5. The Bachelor of Science (B.Sc.) with Computer Science course has the status, as one of the subject at the B.Sc. Course. It is further clarified that adequate laboratory staff (viz. Lab. Assistant and Lab attendant) are required for conduct of B.Sc. Computer Science Practicals, on the same pattern as with the other science subjects such as Physics, Chemistry etc. For smooth conduct of practicals at F.Y., S.Y. & T.Y.B.Sc.(Computer Science), a minimum of 1 Lab. Asst. and a minimum of 1 Lab. attendant be provided for each lab, each session, during entire period of laboratory practical session.

T.Y. B. Sc. (Computer Science, revised) Scheme

	Subject title	No. of Lectures	Theory Marks	Practical Marks	Total Marks
PAPER I Section I Section II	 Systems Software Data Communications and Networking	 48 52	 100	 Group I 50	 150
PAPER II Section I Section II	 Advanced JAVA VISUAL C++	 52 48	 100	 Group II 50	 150
PAPER – III Section I Section II	 Operating Systems UNIX	 52 48	 100	 Group III 50	 150
PAPER – IV Section I Section II	 Structured System analysis and Design Object oriented analysis and design and Software testing	 52 48	 100	 Group IV 50	 150
Applied Component	WEB Design and Technologies and .NET Technologies				
PAPER I	Principles of WEB Design and WEB Technologies.	50	60	Group I (A.C) 40	100
PAPER – II	.NET Technologies	50	60	Group II (A.C) 40	100

NOTE : For students offering “3 units” of computer science at T.Y.B.Sc., following will be the course structure. Paper I & Paper II (of six units) and Group I, Group II practicals (of six unit course) will form the course contents.

T.Y.B.Sc. COMPUTER SCIENCE SYLLABUS (Revised)
Paper I, Section I

CLASS: B. Sc (Computer Science)		**Year III**	
SUBJECT: Systems Software: Paper I Section I Data Communication and Networking: Paper I Section II			
Periods per week (1 Period = 50 minutes)	**Lecture**	**4**	
	Practical	**--**	
		Hours	**Marks**
Evaluation System	**Theory Examination**	**3**	**100**
	Practical per year	**--**	**--**

SYSTEMS SOFTWARE

<u>Unit I</u>

Language Processors - Introduction, Language Processing Activities, Fundamentals of Language Processing & Language Specification, Language Processor Development Tools
Data Structures for Language Processing - Search Data structures, Allocation Data Structures
Scanning & Parsing
Unit References
Code: DMD (Ch 1, 2, 3)

<u>Unit II</u>

Assemblers - Elements of Assembly Language Programming, A Simple Assembly Scheme, Pass Structure of Assemblers, Design of a Two Pass Assembler, A single pass Assembler for IBM PC
Macros and Macro Processors - Macro Definition and Call, Macro Expansion, Nested Macro Calls, Advanced Macro Facilities, Design of a Macro Preprocessor
Linkers - Relocation and Linking Concepts, Design of a Linker, Self-Relocating Programs, A Linker for MS-DOS, Linking for Overlays, Loaders
Software Tools - Software Tools for Program Development, Editors, Debug Monitors, Programming Environments, User Interfaces
Unit References
Code: DMD (Ch 4, 5, 7, 8)

<u>UNIT III</u>
<u>Compilers</u>

Statement of Problem - Recognizing Basic Elements, Recognizing Syntactic Units and Interpreting Meaning, Intermediate form, Arithmetic statements, Non-Arithmetic statements, Non-executable statements, Storage Allocation, Code Generation, Optimization (Machine-independent), Optimization (Machine Dependent), Assembly Phase, General Model of the Compiler
Phases of the Compiler - Lexical Phase, Syntax Phase, Interpretation Phase,
Optimization, Storage Assignment, Code Generation, Assembly Phase, Passes of the Compiler
Data structures - Introduction, Implementation, Recursion, Call & Return Statements,
Storage Classes, Static, Automatic, External Control & Based Storage
Implementation, Block structure, Non-local Go To's, Interrupts, Pointers.
Interpreters - Use & Overview, Pure & Impure Interpreters
Unit References
Code: For Compilers: JD: Chapter 8. Additional Ref: DMD: Chapter 6.
Code: For Interpreters: DMD: Chapter 6.

Main References: 1. DMD: Systems Programming & OS by D.M.DHAMDHERE
(2nd Revised Edition) TMH.
2. JD: Systems Programming by John Donovan TMH.

****

PAPER-I SECTION-II
Data Communication and Networking

Unit IV **(18)**
Introduction - Data Communication, Networks (LAN, MAN, WAN), Protocols (OSI, TCP/IP)
Physical Layer
Signals – Analog Signals, Digital Signals, Analog versus Digital, Data Rate Limits, Transmission Impairment
Digital Transmission – Line Coding, Block Coding, Sampling, Transmission Mode
Analog Transmission – Modulation of Digital Signal, Modulation of Analog Signals
Multiplexing – FDM, WDM, TDM
Transmission Media – Guided Media, Unguided Media
Circuit Switching and Telephone Networks
Unit References
Introduction: Code: CN (Ch 1)
Physical Layer: Code: DCN (Ch3-8)

Unit V **(16)**
Data Link Layer
Error Detection and Correction – Types of Errors, Detection, Error Correction
Data Link Control and Protocols – Flow and Error Control, Stop-And-Wait ARQ, Go-Back-N ARQ, Selective Repeat ARQ, HDLC
Point-to-Point Access: PPP – Point-to-Point Protocol, PPP Stack
Multiple Access – Random Access, Controlled Access, Channelization
Local Area Networks: Ethernet – Traditional Ethernet, Fast Ethernet, Gigabit Ethernet
Connecting LANs – Connecting Devices
Unit References
Data Link Layer: Code: DCN (Ch 10-14, 16)

Unit VI **(16)**
Network Layer
Host-to-Host Delivery - Internetworks, Addressing, Routing
Network Layer Protocol - ARP, Ipv4, and Ipv6
Routing Protocols – Unicast Routing, Unicast Routing Protocol, Multicast Routing, Multicast Routing Protocol
Transport Layer
Process-to-Process Delivery – UDP, TCP
Application Layer
DNS – Name Space, Domain Name Space, Distribution of Name Space, DNS in the Internet
Electronic Mail (SMTP) and FTP
HTTP and WWW
Unit References
Network Layer: Code: DCN (Ch 19-21)
Transport Layer: Code: DCN (Ch 22)
Application Layer: Code: DCN (Ch 25-27)

Main ref: i) DCN - Data Communication and Networking by Behrouz Forouzan (3rd ed)
ii) CN - Computer Networks by Andrew S Tanenbaum (PHI)
Additional References: i) Local area Networks by Keiser G E (TMH)
ii) Data and computer communication by William Stallings PHI (5th ed)

****

CLASS: B. Sc (Computer Science)		**Year III**	
SUBJECT: Advanced Java: Paper II Section I Visual C++: Paper II Section II			
Periods per week (1 Period = 50 minutes)	**Lecture**	**4**	
	Practical	**8 (Gr. I & Gr. III)**	
		Hours	**Marks**
Evaluation System	**Theory Examination**	**3**	**100**
	Practical per year	**3 + 3 (Gr. I & Gr. III)**	**100**

Paper II Section I
Advanced Java

Unit I

Overview of Networking-Networking Basics
Working with URLs-What Is a URL?, Creating a URL, Parsing a URL, Reading Directly from a URL, Connecting to a URL, Reading from and Writing to a URLConnection.
Sockets-What Is a Socket?, Reading from and Writing to a Socket, Writing the Server Side of a Socket.
Datagrams-What Is a Datagram?, Writing a Datagram Client and Server, Broadcasting to Multiple Recipients
Swing Features and Concepts
Using Swing Components-The JComponent Class, **Using Top-Level Containers** (Frames, Dialogs, Applets), **Using Intermediate Swing Containers** (Panels, Scroll Panes, Split Panes, Tabbed Panes, Internal Frames, Layered Panes, Root Panes)
Using Atomic Components (Buttons, Check Boxes, and Radio Buttons, Combo Boxes, Labels, Lists, Menus, Tables, Text Components, Trees)
Laying Out Components Within a Container-Using Layout Managers (BorderLayout, BoxLayout, CardLayout, FlowLayout, GridLayout, GridBagLayout)
Writing Event Listeners - Implementing Listeners for Commonly Handled Events (Action Listener, Component Listener, Container Listener, Focus Listener, Internal Frame Listener, Item Listener, Key Listener, Mouse Listener, Mouse-Motion Listener, Window Listener)
Unit References:

Code: J2CR (Ch 18, 20, 26)

Code: JPAT (Ch 13, 16)

Unit II

JavaBeans Concepts and the Beans Development Kit - JavaBeans Concepts, The Beans Development Kit
Using the BeanBox - Starting and Using the BeanBox, The BeanBox Menus, Using the BeanBox to Generate Applets
Writing a Simple Bean
Properties - Simple Properties, Bound Properties, Constrained Properties, Indexed Properties
Manipulating Events in the BeanBox, The BeanInfo Interface, **Bean Customization, Bean Persistence**, **Using the BeanContext API**
Unit References

Code: J2CR (Ch 25)

Code: JPAT (Ch 15)

Unit III

Java Server side programming – Servlets, Understanding the http protocol, Writing servlets, Servlet API, Writing servlets to receive requests and send responses,
JSP - Server processing of JSP's, Java programs in JSP's, Applying MVC principles using JSP's and JavaBeans.
JDBC API - loading database drivers, establishing a database connection, issuing dynamic SQL statements, processing a ResultSet

Unit References

Code: JPAT (Ch 17)

Code: J2CR (Ch 27)

Main References

1. Code: JPAT.	Java Programming Advanced Topics, Joe Wigglesworth and Paula Lumby, Course Technology (Thomson Learning), (2000).
2. Code: J2CR.	Java 2 - The Complete Reference 3/e, Patrick Naughton and Herbert Schildt, TMH, (1999).

****

Paper II Section II
Visual C++ Programming

Unit IV

Introduction to Visual C++: Visual C++ Development Environment (IDE), Visual Studio, Managing Projects, Visual C++ Compiler: Compiling and Linking, Debugging
Introduction to Windows Programming and MFC – Windows Basics: Structure of Windows Programs, Application Frameworks, Class Hierarchy of the MFC Library, CWinApp and CFrameWnd Classes, Event-Driven Programming, Windows Messages
Message Maps
User Input in Windows – Mouse (Client area and Non-client area Mouse Messages), Using ClassWizard, Invalidating the Client Area, Keyboard (Keyboard Massage Handlers)
Unit References:
Code: PM(Part-1-Ch.1,Ch.2,Ch.3,Ch.4)
Code: PM (Part IV)
Code: YK(Ch-1,Ch-5,Ch-6)

Unit V

Writing Text and Drawing Graphics - Device Contexts, GDI Object Creation and Cleanup, CGdiObject Class: CPaintDC, CClientDC, CWindowDC, CMetaFileDC
Menus and Resources – The CMenu Class, Resources in Windows Programs, Developer Studio Resource, Editors, String Tables, Menus and Command Messages, Update Command UI Messages, Keyboard Accelerators
Controls and Dialogs - Modal and Modeless Dialog Boxes, Resources and Controls, Controls as Child Windows, CDialog Class and Programming a Modal Dialog, Designing Dialogs with Dialog Editor, Using Class Wizard, Dialog Data Transfer, The windows Common Dialogs, Common controls.
Unit References
Code: YK (Ch.4, 10, 11, 12, 13)

Code: PSW (Ch.13)

Unit VI

Toolbars and Status Bars - Command Messages, MFC Control Bar Classes, Toolbars, Idle Time Processing, Tool tips, Status Bars
Document/View Architecture - Using AppWizard, Document and View Classes, Document Templates, SDI and MDI Applications, Document/View Program Structure
File I/O and Database management - Serialization: Making a serializable class, serializing an object. The MFC ODBC Classes-CRecordSet, CRecordView and CDatabase
Unit References
Code: YK(8,9,14)
Code: PSW(Part V: Ch.3,16,17,31).

Main References:

Code: YK: Visual C++ Programming – Y. Kanetkar (BPB)

Code: PM: The Complete Reference VC++ - Pappas, Murray (TMH)

Code: KSW: Programming VC++ - David J. Kruglinski, George Shepherd, and Scot Wingo (WP).

CLASS: B. Sc (Computer Science)		**Year III**	
SUBJECT: Operating System : Paper III Section I **System Study and Unix System Programming: Paper III Section II**			
Periods per week **(1 Period = 50 minutes)**	**Lecture**	**4**	
	Practical	**4 (Gr II)**	
		Hours	**Marks**
Evaluation System	**Theory Examination**	**3**	**100**
	Practical per year	**3 (Gr II)**	**50**

Paper-III Section-I
Operating Systems

<u>Unit I</u>
Introduction:
What is an Operating System?
Types of Operating System: Evolution of OS
Os for Main frame Computer Systems: Batch Systems, What is multiprogramming? Importance of Multiprogramming, Multiprogrammed Systems, Time-sharing Systems,
OS for Multiprocessor Systems: What is multitasking? Distributed Systems: Client Server & Peer-to-Peer Systems. Clustered Systems. Real time OS
Examples of OS including MS DOS, UNIX, LINUX and WINDOWS (NT, 2000, XP)
(SG: Ch.1)
Operating System structure:
Components of OS: Process Management, Main memory Management, Secondary storage Management, File Management, I/O Management.
Operating System Services, Command Interpreter: interface between user and OS.
System calls: Types of system calls
System programs: Different categories of system programmes.
OS structure: Layered approach, Kernel based approach
OS design issues.
(SG: Ch3)
Process Management: What is a Process?, Process states: two state and five state model, processes & resources, concurrent processes, process description, process control block and its role.
Operation on processes: Creation, Termination,
Cooperation among the processes.
Interprocess Communication: Direct & indirect communication, message passing, synchronization, buffering.
Light weight processes: threads, single & multithreaded processes, user and kernel threads, multithreaded models. Threading issues, Creation of threads
Maximizing CPU utilization: Process scheduling, queuing diagram, scheduler: short term and long-term scheduler, scheduling queues. Need for Process switching, context switching, process synchronization, CPU scheduling.
(SG: Ch 4, 5), (WS: Ch 3)

<u>Unit II</u>
Process Synchronization: General structure of a typical process, Critical Section Problem and its solutions, Two and multiple process solutions, Need for Mutual Exclusion, Classifying process interactions, achieving mutual exclusion: Dekker's Algorithm, Peterson's Algorithm and their final correct solution for two processes.
Tools for process synchronization: Semaphores, Binary semaphores, monitors, message passing: their use & implementation for mutual exclusion.
Classical Problems of Process synchronization:
Producer-Consumer problem for infinite and bounded buffers and its bounded buffer solution using semaphore, monitor and messages Reader-writer problem and its solutions with readers' priority and writers' priority, Dinning-Philosophers Problem and its solutions **(Arrange student's seminars on the classical problems)**
Problems arising due to process concurrency: Deadlock and starvation

Characterization of deadlock: conditions, resource allocation graph. Methods of handling deadlocks. Deadlock prevention. Deadlock detection.
Deadlock avoidance: safe and unsafe state, resource allocation algorithm, Banker’s algorithm. Examples: Given a snapshot of a system check whether system is in a safe state. Recovery from deadlock.
CPU Scheduling:
Basic concepts: CPU- I/O burst cycle, scheduler. Preemptive scheduling. Dispatcher. Scheduling criteria. Scheduling Algorithms. Examples related to finding turnaround time, waiting time for a given set of processes. Evaluation criteria for scheduling algorithms.
Unit References:
(SG: Ch 6,7,8), (WS: Ch 4,5)

<u>Unit III</u>
Memory management: Background. Swapping. Contiguous memory allocation. Paging. Segmentation. Segmentation with paging. Virtual memory: Background, demand paging, process creation. Page replacement: Basic scheme, replacement algorithms. Thrashing: its cause and limiting its effect.
File System: File Concept: attributes, operations, types, structure. File access methods: sequential, direct, other. Different directory structure. File system structures. File system implementations. Directory implementations. Allocation methods. Free space management.
I/O System: Overview, I/O hardware: typical bus structure, polling, interrupts, direct memory access. Application I/O Interface.
Disk Scheduling: FCFS, SSTF, SCAN, C-SCAN. Examples related to disk arm movement.
(SG: Ch 9, 10, 11, 12, 13, 14)

****

PAPER-III SECTION-II

SYSTEM STUDY AND UNIX SYSTEM PROGRAMMING

Unit IV

OS Case studies: Overview of WINDOWS (NT, 2000, XP) and detailed study of UNIX and LINUX in particular (History, Type of the System: Multi-user and multitasking, System calls, System structure, Shell, Kernel, designing principles, processes management: Process, Process table, Child process, Mode of execution, Pipes, Semaphores, threads. File System &management: System directories, directory tree structure. File subsystem, File descriptor, File table, Inode, File Descriptor table, Inode table. Memory management, I/O System and management,)
(Arrange student's seminars on the Example Systems)
(SG: Ch 20,21,22 other chapters)
(WS: Chapters, which describes the example Systems)(YK: Ch1, 3)
Working in UNIX/LINUX part - I: Accessing the system using Graphical User Interface, Features of UNIX/LINUX**:** normal user and super user, logging in and logging out of the system. Understanding UNIX/ LINUX commands. General-purpose utilities. Navigating file system. Handling Ordinary files. The Shell. The vi Editor and its environment. Basic file attributes. Simple filters. Regular expressions and the grep family. I/O redirection and piping.
(SD: Ch1 to12),(YK: related chapters)

Unit V

Working in UNIX/LINUX part - II:
The processes. More file attributes. Links to the file. Communication between the systems.
(SD: Ch13, 14,16) (YK: related chapters)
Security in Unix: Password, Characteristic of good password, Files permissions, Directory permissions. File ownership and groups.
UNIX Shell Programming:
Shell variables and key words. Shell meta characters and their execution. Shell programming: Writing and executing Shell script, read and echo command, other commands and statements: if….then…fi, test, nested if….else…, elif, looping statements.
UNIX C/ C++ Programming
Introduction to compilers gcc, g++. C Shell program. Compiling and Executing C/C++ – programs on UNIX platform.
(SD: Ch15) (YK: related chapters)

Unit VI

Advanced filters: Working with **sed and awk**
(SD: Ch20) **(YK: related chapters)**
UNIX System Administration-I:
System administrator, logging in and logging out as a **root**. Privileges. Operations, Find, dd, handling DOS, backups, copy input-output, tar. Mounting of floppies and CDROMS.
TCP/IP Networking and Internet: Basic features, addresses and addressing system, telnet, ftp, r-utilities, uucp, cu.
UNIX System Administration-II:
Understanding the file system, data blocks, block addressing scheme, partitions: fdisk, file system mounting, checking of file system, adding a hard disk, creating a file system, adding a file system. Creating, deleting, assisting and managing users. Startup and shutdown: init, init Run levels. Managing the terminals, printers. Monitoring system usage. Ensuring system security.
Unit References
(SD: Ch17, 18,19,25,26) (YK: related chapters)

Main References:
Section I
1. Code:SG: Opearting System Concepts, Silberschatz, Galvin, Gange (sixth edition)
2. Code:WS: Operating Systems (Second Edition) by William Stallings. (PHI). Section II:
3. Code SD:UNIX System and concepts (latest edition), by Sumitabha Das (TMH)
4. Code:YK: UNIX Shell programming by Yashwant Kanetkar (BPB)
5. Code:JM:UNIX made easy by John Muster (TMH)

Additional References:
Operating Systems Design and Implementation, Andrew S Tanenbaum, Albert S Woodhull, (Prentice-Hall)
Operating Systems with case studies in UNIX NETWARE, WINDOWS NT : Achyut S Godbole, (TMH)
Working with Unix by Vijay Mukhi , (BPB)
UNIX – The Complete Book, A guide for professional Users (Galgotia)
Understanding Unix – A conceptual Guide, R.Groff & P.N. Weinberg (BPB)
The UNIX Programming environment , Pike rob & Kerningham Brain W, (Prentice Hall)

****

CLASS: B. Sc (Computer Science)		**Year III**	
SUBJECT: SSAD: Paper IV Section I **OOAD and Software Testing: Paper IV Section II**			
Periods per week **(1 Period = 50 minutes)**	**Lecture**	**4**	
	Practical	**4**	
		Hours	**Marks**
Evaluation System	**Theory Examination**	**3**	**100**
	Practical per year	**4**	**50**

Paper IV Section I
Structured Systems Analysis & Design

Unit I

What is a System? - The general systems approach to problem solving. The three approaches to software systems development - The Structured approach, the Object Oriented Approach and the Information Engineering Approach.

System Development Projects and the SDLC -The Systems Development Life Cycle, The First four phases of SDLC, Scheduling the project phases

Software Development Life Cycle Models - Waterfall Model, Prototyping Model, RAD Model, Incremental Model, Spiral Model, Concurrent Development Model, Component Based Model, Formal Methods Model and Fourth Generation Techniques. – Their features, strengths, weaknesses and differences between them.

Project Feasibility Study - operational, technical, economic, organizational and cultural feasibility. Defining project costs and project benefits. Cost/Benefit Analysis for a project – Net present value, payback period and return on investment computations.

Unit References:
1. Code: SADCW. (Ch. 2,3).
2. Code: SEPA. (Ch. 2,11).
3. Code: SADM. (Ch. 2,4, Module B).

Unit II

Investigating System Requirements - Functional and Technical Requirements, The sources of system requirements, identifying system requirements (Sampling documents, forms and files. Site visits, Observation of Work environment. Questionnaire formulation, Interviewing techniques), structured walkthroughs

Modeling System Requirements - the purpose, type and overview of models. Modeling system requirements for events. Modeling system requirement for objects, roles, devices, organizational units, and locations

Data Modeling - Data entities attributes and relationships, the Entity-Relationship diagram

Process Modeling - Developing Data Flow Diagrams, Level of abstraction, Context diagram, Top level DFD, DFD fragments, The event-partitioned system model, Decomposing processes, Physical and Logical DFD, Evaluating DFD quality, Documenting DFD components, The concept of data dictionary, Process, data flow, data store, data elements descriptions

Representing Process Logic - building decision tables, decision trees, structured English, tight English and pseudocode, their usage and differences

Unit References:
1. Code: SADCW. (Ch. 4,5,6).
2. Code: SEPA. (Ch. 12).
3. Code: SSAIT. (Ch. 5).

Unit III

Application Architecture Design - Determining the automation system boundary

Software Design - Designing the system flowchart and the system level structure chart. Transaction analysis and transform analysis.

Designing Databases - Databases and DBMS Designing Relational DBMS's. Normal forms upto 3rd normal form. Representing entities, relationships, enforcing integrity constraints and business rules.

Designing system inputs outputs and controls

Designing the user interface - Interface design guidelines. Dialog design. Designing Windows forms.

Unit References:
1. Code: SADCW. (Ch. 9,10,11,12).
2. Code: SEPA. (Ch. 13,15).

Paper IV Section II
Object Oriented Systems Analysis Design & Software Testing

Unit IV

Object Oriented Requirements Specifications and Analysis - the Unified Modeling Language, the Case diagrams, class diagrams, object diagrams, Collaboration and sequence diagrams, states, state transitions and state chart diagrams,

Designing the Application Architecture-The Object Oriented Approach - activity diagrams, component diagrams, deployment diagrams, Package diagram

Object Oriented Databases - Designing object databases, representing classes and relationships

Hybrid Object-Relational Databases - Classes and attributes. Relationships. Relational DBMS and object DBMS data types

Unit References:
1. Code: SADCW. (Ch. 7, 9,10).
2. Code: IUML. (Ch. 3).

Unit V

Software Testing Techniques – Software Testing Fundamentals, White Box Testing, Basic Path Testing, Control Structure Testing, Black Box Testing, Testing for Specialized Environments, Architectures, and Applications
Software Testing Strategies – A Strategic Approach to Software Testing, Unit Testing, Integration Testing, Validation Testing, System Testing
Unit References:
Code: SEPA (Ch 17,18)

Unit VI

Object Oriented Testing – Testing OOA and OOD Models, Object Oriented Testing Strategies, Test Case Design for Object Oriented Software, Testing Methods Applicable at the Class Level, Interclass Test Case Design
Unit Reference:
Code: SEPA (Ch 23)

Main References:

1. Code: SADCW.	Systems Analysis and Design in a Changing World, Satzinger, Jackson and Burd - Thomson Learning/Course Technology, (2000).
2. Code: SEPA.	Software Engineering – A Practitioner's Approach 5/e, Roger S. Pressman – McGraw-Hill International Edition (2001).
3. Code: SADM.	Systems Analysis and Design Methods, Whitten and Bentley – Tata McGraw Hill (1998).
4. Code: IUML.	Instant UML, Pierre-Alain Muller – Wrox/SPD (1997).

Additional References:

1. Code: SSATT.	Structured Systems Analysis: Tools and Techniques, Gane and Sarson – Prentiss Hall (1979).
2. Code: UMLN.	UML in a Nutshell, Sinan Si Alhir – Wrox/SPD (1998).

Practicals, Group I
Advanced Java - Practicals

Notes:

The following instructions about Java practicals should be carried out in the interest of uniformity in their conduct across all the constituent colleges of University of Mumbai.

1. The practicals should be performed on the J2SE jdk 1.3 platform. The J2SE jdk 1.3 is downloadable from the java.sun.com website.
2. The compilation and execution should be from the bare command prompt available on the Windows 9.x or compatible system. Other IDE's should not be used for program entry, editing, debugging, compilation or testing.
3. Methods available in jdk 1.0, 1.1 and 1.2, which are deprecated by jdk 1.1, 1.2 or 1.3 should not be used in programs.
4. The practical problems must be solved using the best possible methods available on the jdk 1.3, irrespective of whether those methods form a part of the T.Y.B.Sc. (Theory) Advanced Java-I and/or Advanced Java-II syllabus or not. Though there is a substantial correlation between the T.Y.B.Sc. Java theory and practicals, the practicals are not to be treated as a subset of the theory.
5. In each program, the input data (where it exists), should be output along with the results. The numeric values output should be preceded by explanatory text.
6. Exceptions should be generated and exception-handling code should be written wherever necessary.
7. Some of the programs could become too long to complete in one lab session. The instructor may, in that case supply a part of the program and ask the student to complete it. The partly complete program could also contain deliberate errors, so that the student debugs the program after completing its coding. The errors could be syntactic as well as logical.
8. Classes should be properly encapsulated, so that the data structures are not directly accessible from outside of classes.
9. Polymorphism and inheritance should be implemented wherever necessary.

PRACTICALS

1 Define a class to download a file from the Internet and either copy it as a file on the local machine, or output it to the screen.

2 Define a class that displays information about a file URL like its type, encoding, length, dates of creation, last modification and expiry. Additionally the class should display the request method, response message and the response code for a Web URL.

3 Build an applet that sends a datagram to a server each time that it is started or stopped. Further build a server application that receives the sent datagram and prints it out.

4 Define a class that enables the drawing of freehand lines on a screen through mouse clicking and dragging. Use anonymous inner classes to implement event listeners. The drawing should be cleared when a key is pressed and the line colour should be selectable. Define a test class to demonstrate the program.

5 Write a program to create a form to enter biodata of student. Use various components such as JLabel, JButton, JTextField, JTextArea, JCombobox, JTable, JScrollPane, JOptionButton, JCheckBox. Use GridBagLayout and GridBagConstraints to lay the components.

6 (Creating MDI Form in SWING)Design a class, which extends JFrame and has three objects of three classes whose name are ColorPanel, CitiesPanel, and FlavourPanel. All the three classes (ColorPanel: Names of colors inside it, CitiesPanel: Names of cities inside it, and FlavourPanel- Names of Icecream flavours) extend JInternalFrame and have a method, which makes the Panel visible.

The main JFrame has an object of JDesktopPane and this object holds the three panels, which are displayed on the click of the menu bar item.

7 Create a form using JApplet having 2 fields namely Name and Remarks. Also create a button called Insert. This button will insert these data from the form in a text file and later on read the same data and print it. If more number of records are entered it should be appended to the same file. Assume that the text file is already present.
Also notify the Exceptions thrown by the above program and give appropriate messages there.

8 Create an animation with a single line, which changes its position in a clock-wise direction. This line should produce an effect of a spinning line. In the same example give option for clock-wise or anti-clock-wise spinning. Also provide start and stop buttons to start and stop the animation. Demonstrate the animation on the screen.

9 Develop a Java bean that is a 2D shape filled with colour, which could take on different shapes. The colour changes randomly each time the mouse is clicked on the bean.

10 Create a Java bean that displays multiple lines of text with margins and alignment.

11 Create a Java bean that accepts numeric data as text, validates it as a number and then makes the numeric value available for further processing.

12 Create an invisible Java bean that multiplies the values of two of its properties and places the product in a third as a read only property.

13 Write a servlet that accepts text and numbers sent from an HTML document and displays them on screen.

14 Write a servlet that accepts single-valued as well as multi-valued parameters like check boxes and multiple selection list boxes from an HTML document and outputs them to the screen.

15 Write a servlet that reads the name of the servlet class and its initialisation parameters from a server file and then displays these parameters on the screen.

16 Write a servlet that redirects the user to a different Web page when an out of sequence Web page is requested by him/her

Lab Programs

Perform at least Six *practicals from programs 1-8 and at least* Six *from practicals from programs 9-16.*

Reference:

Java Examples in a Nutshell – David Flannagan, SPD-O'Reilley (2000).

****

GROUP: II

VISUAL BASIC, LINUX & SHELL Programing Practicals
VISUAL BASIC 6.x

1. Accessing Database files:
    a) Databases supported by VB 6.
    b) Creating database files for Use by VB
    c) Using the Data Control.
    d) Viewing the database file.
2. Navigating the Database:
    a) Using List Boxes and Combo boxes as Data-Bound Controls.
    b) Adding a Look up table and Navigation.
    c) Updating a Database File, Preventing Errors.
3. Advanced data handling-FlexGrids, Validation, Selection and SQL:
    a) Displaying Data in Grids.
    b) Validation and error trapping
    c) Validation Techniques.
    d) Recordsets, Searching.
    e) Reordering a Table Recordset
    f) Creating a New Dynaset Using SQL.
    g) Tracking database Errors.
4. ActiveX:
    a) Using ActiveX controls
    b) The tabbed dialog control
    c) Browsing the web form.
5. Generating crystal reports
6. Creating a Simple Chat Application using WinSock.

Visual Basic .NET

7. Creating windows form and Creating a class in VB.NET
8. Implementing Inheritance and Creating Procedures in VB.NET
9. Connecting to a database
10. Implementing simple data binding and Implementing complex data binding
11. Filtering and sorting data.
12. Displaying data from multiple tables

NOTE: 1. Some of the experiments listed above may require more than one turn. The teacher should decide and allot the time required for each title listed above
2. Minimum number of experiments required is **TEN** from above.
3. Reference for above practicals:

Programming in VB 6.0 by Julia Bradly & Anita Millspaugh (TMH)
Beginning VB.NET 2003 – Thearon Willis, et al. (WROX)

LINUX & SHELL PROGRAMMING:
(The Practicals can be performed in LINUX)

1. LINUX Installation (Demo Practical)
2. Working in LINUX – Introduction to GUI and Command line interface, Understanding File System, Procedure for logging in and out, creation of user account.
3. General Purpose Utilities, directory related commands
4. Basic LINUX file related commands: touch, cat, cp, rm, ls,
5. Study of Commands: more, find, tr, head, tail, wc, file, sort, and split, ln.
6. Study of filter commands: od, cmp, comm, diff, uniq.
7. Study of Commands: grep, egrep, fgrep.
8. Advance Shell programming I
9. Advance Shell programming II
10. Editors In Linux: vi, ex, sed, awk
11. System Administration I:

12. System Administration II:

Minimum **six** practicals from above should be completed.

References 1. UNIX SysstemV.4 Concepts & Applications By Sumitabha Das (TMH)
2. Using Linux by Bill Ball (Que-PHI)

****

Group III

Visual C++ - Practicals

Lab Programs

1) Write a program to print the “Hello World” in client area
2) Write a program to count the number of right click, left click, right double click, left double click.
3) Write a program to draw different types of shape (line, rectangle, ellipse, round rectangle, pie, polygon) using different styles of pen.
4) Write a program to fill the random color in rectangle when u will press Ctrl + C keys.
5) Write a program to fill the rectangle using different types of brushes.
6) Write a program to display a bitmap on client area.
7) Write a program to create a color menu. And on selection of color the rectangle should fill with respective color.
8) Write a program to create a shape toolbar. Display the selected shape on the client area.
9) Write a program to create a status bar & show the status of Caps lock, Num lock keys, & also show the status of current mouse coordinates.
10) Write a program to enter the user name & password on dialog box. When you will select OK button then entered information is display on client area.

11) Write a program to save the record in the file. Read the recorded message and display it on a message box.

12) Write a program to display the records in access database.

12) Write a program to create a notepad application.

13) Write a program to create a paintbrush application.

14) Write a program to create a calculator application.

Note: Perform any 12 practicals from the list.

****

Group IV: Practicals

1. Seminars, presentations and quizzes:
Given and arranged by the students on the topics in the syllabus and topics on new, current developments in the software industry. They should arranged either in the lab or in a classroom which ever is preferable, whenever there are no labsessions related to project.

2. Group IV Project:
Lab session 1: Choosing a topic for project. Basic components of project. Time management, Schedule for the project.

Lab Session 2: Demo of a complete project

General guidelines for project performance evaluation:

The student is required to analyze design, code and implement an actual system. The system could be implemented either as a traditional system, an event driven system, object-oriented system or as a database system. The coding and implementation should be in one of the languages/packages that are defined in the Computer Science syllabus of either F.Y.B.Sc., S.Y.B.Sc. or T.Y.B.Sc.

Each student shall do the project individually, though a project with the same title could be given to more than one student.

A project guide should be assigned to students, who will generate a schedule for the completion of each of the phases of the project and hand it over to the students under his care before the 15th of July in the third year of the Computer Science course. ***It is highly recommended that the colleges try and obtain the services for external project guides who have working experience of managing projects***. The guides should oversee the project progress on a weekly or fortnightly basis.

The student will produce and maintain a Gantt chart from the given schedule. The actual completion of each phase should also be noted on the chart. The chart shall be placed in the project documentation immediately after the Certificate and separate marks are assigned for completing each phase on time, as indicated by the Gantt chart during project evaluation.

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

The phases could be suitably modified if the waterfall paradigm is not used for project development or if the project uses Object Oriented techniques. The project guide should control iterations if any non-linear technique for project development, like prototyping, is used. The design phase must finish in the first term so that coding and unit testing could commence during the Diwali vacation. The project must be completed latest by 31st January of the academic year.

The project must be periodically reviewed by the project guide, who should certify the completion of each phase by affixing his signature on a table maintained for the purpose. The date of the signature assumes importance in view of the marks assigned in evaluation for completing each phase on time. A phase is deemed completed only when its documentation is submitted and approved. The signed sheet should appear in the project report after the Gantt cart.

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.

The completed project must be reviewed in the in the month of February (or earlier) by the project guide to ensure that it is correctly working. The project guide should affix his signature on the certificate after the final review is conducted.

The certificate should minimally contain the following information.

- The fact that the student has successfully completed the project as per the syllabus, and that it forms a part of the requirements for completing the B.Sc. degree in Computer Science of University of Mumbai.
- The names of the student and the project guide.
- The academic year in which the project is done.
- Date of submission and final review of the project.
- Signature of the project guide and the head of department with date along with the department stamp.
- Space for the signature of the University examiner and date on which the project is evaluated.

  The project report should contain the following system documentation (one copy for the project).

- Organizational Overview
- 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.

The project report should contain the following program documentation (one copy for each program).

- Program id.
- Program level run chart.
- Program function explanation.
- Data entry screen (reproduced from system documentation).
- Report layout (reproduced from system documentation).
- Program level pseudocode or flowchart. Add decision tables, decision trees, tight English explanation where necessary.
- Program listing.
- Test data.
- Test results.

The University examiner along with the internal guide shall evaluate the project at the time of University practical examinations. The system code should be available on a disk for evaluation along with the project report. The examiner should critically evaluate whether the project has been sufficiently tested by going through the test data and test results.

The student is expected to present and demonstrate the project to the University examiner.

50 marks are allotted to the project, which shall be divided as under.

1. **Adherences to project schedule : 25 Marks.**
   & Internal demos and presentation
   (Should be evaluated by the internal guide on the basis of the yealy performance of the student)

2.

Project documentation	**: 5 Marks.**
Project quality	**:10 Marks.**
Demonstration of working project	**: 5 Marks.**
Project presentation	**: 5 Marks.**
Total	**: 25 Marks**

(Should be evaluated by the External faculty member)

The examiner may deduct 2 marks, up to a maximum of 10, from the 'Adherence to project schedule' head for each schedule milestone that is missed by the student. The examiner may deduct 2 marks from the 'Project quality' head if the test data and results are not sufficiently exhaustive.

****

T.Y.B.Sc. COMPUTER SCIENCE
APPLIED COMPONENT

PRINCIPLES OF WEB DESIGN & WEB TECHNOLOGIES AND .NET TECHNOLOGIES

CLASS: B. Sc (Computer Science)		**Year III**	
SUBJECT: Principles Of Web Design & Web Technologies: Applied Component - I			
Periods per week (1 Period = 50 minutes)	**Lecture**	**2**	
	Practical	**2**	
		Hours	**Marks**
Evaluation System	**Theory Examination**	**3**	**60**
	Practical per year	**3**	**40**

PAPER-I

PRINCIPLES OF WEB DESIGN & WEB TECHNOLOGIES

Unit I
Web Site Design Principles – Design for the Medium, Design for the Whole Site, Design for the User, Design for the Screen
Planning the Site – Create a Site Specification, Identify the Content Goal, Analyze your Audience, Build a Web Site Development Team, Filenames and URLs, Directory Structure, Diagram the Site
Planning Site Navigation – Creating Usable Navigation, Using Text-Based Navigation, Using Graphics-Based Navigation
Creating Page Templates – Understanding Table Basics, Table Pointers, Creating a Page Template

Unit References
Code: PWD (Ch 2,3,4,5)

Unit II
Web Typography – Type Design Principles, Controlling Typography with the <FONT> Element, Controlling Typography with Cascading Style Sheet, Styling with CSS
Graphics and Color – File Format Basics, Computer Color Basic, Choosing a Graphics Tool, Using the <IMG> Element, Working with Hexadecimal Colors
HTML Frames – Understanding Frames, Frame Syntax, Targeting in Framesets, Planning Frame Content
Publishing and Maintaining Your Web Site – Publishing Your Web Site, Testing Your Web Site, Refining and Updating Your Content, Attracting Notice to Your Web Site

Unit References
Code: PWD (Ch 6,7,8,9)

Unit III
HTML - HTML 4.0 Tag Reference, Global Attributes, Event Handlers, Document Structure Tags, Formatting Tags, List Tags, Hyperlinks, Image & Image map, Table Tags, Form Tags, Frame Tags, Executable Content Tags and Style Sheets
Unit References
Code: ELJO (Ch 3-9)

Unit IV
Introduction to Java Script (Functions, Objects)
Client-Side Java Script
Java script in web browser, windows and frames, the document object model, events and event handling, forms and form elements, dynamic html and saving state with cookies

Unit References
Code: JSDG (Ch 7, 8, 11-18)

Unit V

Active Server Pages - Server side Scripting and Client-side scripting, ASP Objects, Scripting Objects (creating an instance of scripting object, The FileSystemObject object [drive, folder and file]) Active Data Object (Connection and Recordeset)

Unit References
Code: B-ASP (Ch 2, 7, 8, 10, 12, 13)

Unit VI

XML - Introduction to XML, Problems with HTML & SGML, Types of XML Markup, Document Type Definitions, Linking, Using Style Sheets with XML, Creating XML Documents.

Unit References
Code: ELJO (Ch 12, 13)

References:
1. ELJO: Using HTML 4, XML & JAVA by Eric Ladd & Jim O'Donnell. (Platinum Edition) (PHI)
2. PWD: Principles of Web Design by Joel Sklar
3. JSDG: Java Script the definitive guide by David Flanagan
4. B-ASP: Beginners ASP 3.0 David Buser et. al. Wrox Press Ltd.

PAPER-II

CLASS: B. Sc (Computer Science)		**Year III**	
SUBJECT: .NET TECHNOLOGIES: Applied Component - II			
Periods per week (1 Period = 50 minutes)	**Lecture**	**2**	
	Practical	**2**	
		Hours	**Marks**
Evaluation System	**Theory Examination**	**3**	**60**
	Practical per year	**3**	**40**

.NET Framework ,VB.NET & ASP.NET

Unit I .NET Framework

Introduction to .NET: Goals of .NET, Building blocks of .NET: .NET Framework, .NET Enterprise Servers, .NET Building Block Services, Overview of .NET Applications
Overview of .NET Framework: Highlights of the .NET Framework, Design goals of the .NET Framework, The Architecture of .NET Framework, The Common Type System, Meta Data, The Common Language Specification
Common Language Runtime: Design goals of CLR, Overview of CLR, The .NET Class Framework
Unit References
PNF (Ch 1, 2)

Unit II .NET Framework

Memory Management Under the CLR: Common Runtime System, Data Storage, Managed Heap Organization, Managed, Unmanaged and Unsafe, Garbage Collection, Garbage Collection Algorithm, Finalize
Working with Runtime: What is MSIL?, CLR, What is an Assembly?, Different types of Assemblies, Common Type Systems, Meta Data, CLS, Reflection API
System Classes: Applications of the System Namespaces – WinCV Tool, String Handling, and Collection Classes
Unit References
PNF (Ch 3, 4, 6)

Unit III VB.NET

Introduction: Information and Data, Variables, Comments and Whitespace, Data Types, Storing Variables, Methods
Controlling the flow: Making Decisions, The if Statement, Select Case, Loops
Working with Data Structures: Arrays, Enumeration, Constants, Structures, Collections, Lists, Building Lookup Tables with Hashtable
Building Windows Application: Responding to Events, Building a Simple Applications, Creating Complex Application, Using Multiple Forms
Displaying Dialog Boxes: MessageBox, OpenDialog, SaveDialog, FontDialog, ColorDialog, PrintDialog
Creating Menus: Understanding Menu Features, Creating Menus, Context Menus
Unit References
BVB.NET (Ch 3-8)

Unit IV VB.NET

Building Objects: Understanding Objects, Building Classes, Reusability, Constructors, Inheritance, The Framework classes
Advanced OO Techniques: Building a Favorites Viewer, Using Shared Properties and Methods, Understanding OOP and Memory Management
Building Class Libraries: Understanding Class Libraries, Using Strong Names, Registering Assemblies, Designing Class Libraries

Creating Your Own Custom Controls: Windows Forms Controls, Exposing Properties from User Controls, Inheriting Control Behavior, Design Time or Runtime, Creating a Form Library
Accessing Databases: Data Access Components, Data Binding
Database Programming: ADO.NET, The ADO.NET Classes in Action, Data Binding
Unit References
BVBNET (Ch 10-13,15,16)

<u>Unit V ASP.NET</u>
ASP.NET Applications: ASP.NET Applications, Code-Behind, The Global asax Application File, Understanding the ASP.NET Classes, ASP.NET
Web Form Fundamentals: A Simple Page Applet, Improving the Currency Converter, HTML Control Classes, The Page Class
Web Control: Stepping Up to Web Controls, Web Control Classes, AutoPostBack and Web Control Events, Assessing Web Controls
Validating and Rich Controls: The Calendar Control, The AdRotator, The Advertisement File, The AdRotator Class, Validations, Understanding Regular Expressions
Unit References
CRASPNET (Ch 5, 6, 7, 9)

<u>Unit VI ASP.NET</u>
Overview of ADO.NET: Introducing ADO.NET and Data Management, Characteristics of ADO.NET, The ADO.NET Object Model
ADO.NET Data Access: SQL Basics, Accessing Data, Creating a Connection, Defining a Select Command, Using a Command with a DataReader, Updating Data, Accessing Disconnected Data, Selecting Multiples Tables
Data Binding: Introduction, Single-Value Data Binding, Repeated-Value Data Binding, Data Binding with Databases
DataList, DataGrid, and Repeater: Introducing Templates, Using Templates with the DataList, Data Binding with Multiple Templates, Comparing the Template Controls, Selecting Items, Editing Items, Paging and Sorting with DataGrid
Using XML: XML's Hidden Role in .NET, The XML Classes, XML Validation, XML Display and Transforms, XML in ADO.NET
Unit References
CRASPNET (Ch 12, 13, 14, 15, 17)
Main References:
PNF: Professional .NET Framework – Kevin Hoffman and Jeff Gabriel (SPD)
BVBNET: Beginning VB.NET 2003 – Thearon Willis, et al. (WROX)
CRASPNET: The Complete Reference ASP.NET – Matthew MacDonald (TMH)

T.Y.B.Sc. COMPUTER SCIENCE
APPLIED COMPONENT PRACTICALS

GROUP I (A.C.)

PRACTICAL LIST (HTML, JavaScript, ASP, DHTML)

1 Basic HTML Tags
- To study Document Structure Tags
- To study Text formatting Tags
- To study Phrase formatting Tags
- To study Listing Tags

2 Advanced HTML Tags
- Table tags
- Merging of tables

3 To Study Form tags and Frames tags

4 Client Side Image Mapping

Cascading Style Sheet

5 Using Editor Dreamweaver
- Creating College home page

6 XML using CSS
- Internal DTD
- External DTD

7 **Form Validation Using Java Script**
- Window Object
- Location Object
- History Object

8 ASP
- Registration Form
- Creating Friends Details Diary

9 ASP
- Server Object (To create a text file at server)
- To maintain total number of visit to an application

10 Dynamic HTML

ASP.NET

11 **Creating a Microsoft ASP.NET Web Form**
- Creating the default.aspx Web Form
- Creating the life.aspx Web Form

12 **Adding Functionality to a Web Application**
- Creating a **Page_Load** Event Procedure
- Creating a **Click** Event Procedure

13 **Validating User Input**
- Using the RequiredFieldValidator Controls
- Using the ValidationSummary Control
- Using the CompareValidator Control
- Using the RegularExpressionValidator Control

14 **Accessing Data with Microsoft ADO.NET**
- Using a SqlDataReader
- Viewing Data from the Database

15 **Reading XML Data**
- Reading a List from an XML File
- Reading, Transforming, and Displaying XML
- Nested Data

All practicals from above should be performed.

GROUP II (A.C.)

WEB PROJECT

Important Note: The Web development is as much about creativity as about technology. While teaching (as well as while evaluating the student performance) proper emphasis should be given to each of the two aspects. The content creation is as much important as the technical know-how. All the time, case studies based on good sites should be given as illustrative examples while teaching this subject.
The student is expected to present and demonstrate the web project to the University examiner.
40 marks are allotted to the project, which shall be divided as under.

1.	**Adherences to project schedule & Internal demos and presentation**	**: 20 Marks.**
	(Should be evaluated by the internal faculty member)	
2.		
	Project documentation	**: 5 Marks.**
	Project quality	**: 5 Marks.**
	Demonstration of working project	**: 5 Marks.**
	Project presentation	**: 5 Marks.**
	Total	**: 20 Marks**

(Should be evaluated by the External faculty member)

Case Study: To Conceive, design & Implement a complete and integrated Web-site of one existing or hypothetical organization. (Upload this web-site on a server) Prepare a project Report & Discuss the designing and issues related to the development of this web site.

Note: 1. The teacher should encourage the students to get a live project for this purpose for as many students as possible.
There should be external evaluation (Viva) of this project.
The project (Web-Site) should be uploaded on local & remote servers & be shown **on-line** to the examiner for evaluation as far as possible. (On-line evaluation is advisable to check the cross platform compatibility and other aspects of the project).

References:

1. The Web Programming Desktop Reference (6 in 1) (Que-PHI)
2. HTML by Xavier (TMH)
3. Online documentation from Microsoft, Sun Microsystems, Netscape etc.

****