

Bachelor of Science – Information Technology

FLEXILEARN

-Freedom to design your degree



Programme Structure

Curriculum & Scheme of Examination

2015

AMITY UNIVERSITY CHHATTISGARH

RAIPUR

B.Sc. – Information Technology

Programme Structure

FIFTH SEMESTER

Course Code	Course Title	Lecture (L) Hours Per Week	Tutorial (T) Hours Per Week	Practical (P) Hours Per Week	Total Credits
IFT2501	Data Communication & Computer Networks	2	-	-	2
IFT2502	Java Programming	3	-	-	3
IFT2503	Designing & Supporting Computer Network	2	-	-	2
IFT2504	Java Programming Lab	-	-	2	1
IFT2505	Designing & Supporting Computer Network Lab	-	-	2	1
IFT2535	Summer Project – II (Evaluation)	-	-	-	3
Concentration Electives					3
IFT2506	Basics of OS (UNIX, LINUX & Shell Programming)	2	-	-	2
IFT2507	Basics of OS (UNIX, LINUX & Shell Programming) Lab	-	-	2	1
IFT2508	Emerging Technologies	3	-	-	3
IFT2509	Software Engineering	2	1	-	3
Open Electives					4*+3
CSS2551	Employability Skills*	1	-	-	1
BEH2551	Individual Society and Nations*	1	-	-	1
LAN2551	Foreign Language – V*	2	-	-	2
LAN2552	French- V				
LAN2553	German- V				
LAN2554	Spanish- V				
LAN2555	Russian - V				
LAN2556	Chinese – V				
LAN2557	Portuguese –V				
LAN2558	Korean-V				
	Japanese-V				
TOTAL					22

Syllabus – Fifth Semester

DATA COMMUNICATION AND COMPUTER NETWORKS

Course Code: IFT2501

Credit Units: 02

Course Objective:

The objective of the course is to provide introductory concepts in Communications such as Signaling, Encoding, Modulation, Error Detection & Correction.

The course is also aimed at providing basic understanding of Computer networks starting with OSI Reference Model, Protocols at different layers with special emphasis on IP, TCP & UDP and Routing algorithms.

Course Contents:

Module I

Introduction to Data Communication, Networks-protocols, advantages, disadvantages & applications, Line Configuration, topology, Transmission mode, Classification of networks.

Parallel & Serial Transmissions, Analog & Digital Signals, Periodic & Aperiodic Signals,

Module II: Digitization techniques

Analog to Analog conversion, comes under Analog to Digital conversion, Digital to Analog conversion.

Module III

Bit/ Baud Comparison, DTE-DCE Interface.

OSI Model, Transmission Media-Twisted Pair Cable, Coaxial Cable, Fiber-Optics Cable, Radio frequency Allocation, Terrestrial Microwave, Infrared rays, Satellite Communication, Cellular Telephony. Introduction to ISDN.

Module IV

Framing, Line Discipline, Types of Errors, Error Detection & Correction (VRC, LRC, CRC, Checksum, Hamming Code), Flow Control (Stop-and-wait & Sliding Window), Error Control (Stop & Wait ARQ, Sliding Window ARQ using Go-back n method and Selective-Reject).

CSMA/CD, Project 802, IEEE Standards-802.3, Token Bus (802.4), Token Ring (802.5), Introduction to Bridges.

Module V

Internal Organization of Network Layer, Routing Algorithms-Shortest Path Routing, Flooding, Distance Vector Routing, Link State Routing,

Module VI

Introduction to TCP/IP, Data Link Layer in Internet-SLIP & PPP, Network Layer in Internet-IP protocol, IP addressing, Subnetting, Transport Layer in Internet-TCP & UDP.

Examination Scheme:

Components	CT1	A/C/Q	Attd	EE
Weightage (%)	10	15	5	70

Text & References:

Text:

- Behrouz ., Forouzan., “Data Communication and Networking”, TMH

□ A.S. Tanenbaum, "Computer Networks", PHI

References:

□ W.Stallings, "Data and Computer Communication" PHI

- Kennedy, “Electronics Communication System”, TMH

JAVA PROGRAMMING

Course Code: IFT2502

Credit Units: 03

Course Objective:

The objective of the course is to give proficiency in developing applications and applets in Java, in-depth knowledge of object oriented concepts, developing GUI applications in Java, creation of packages, Event Handling, Layout Manager, SWING and String handling in Java

Course Contents:

Module I: Introduction

Concepts of OOP, Features of Java, How Java is different from C++, Data types, Control Statements, identifiers, arrays, operators. Variables, Applications and Applets

Module II: Java Programming

Classes and methods, Constructor, method overloading

Inheritance: method overriding, Abstract classes, Final classes, String classes

Module III: Java Packages and interface

Package creation, Packages deployment, using packages, Interfaces

Module IV: Exception Handling and Multithreading

Exception handling, fundamentals exception types, uncaught exceptions, throws, throw, try and catch, final, built in exception, creating your own exception

Multithreading fundamentals, Creating, Implementing and Extending thread, thread priorities,

Module V: AWT and Event Handling

Introduction to AWT, Layout Manager, Event handling Mechanism, Event Model, Event Classes, Sources of Events, Event Listener Interfaces

AWT: Working with Windows, AWT Controls

Html basic tags, Applet Classes, Swing classes & controls. Advantages of Swing over AWT.

Examination Scheme:

Components	CT1	PR.	ATTD.	EE
Weightage (%)	10	15	5	70

Text & References:

Text:

- Naughton, Schidt “The Complete Reference JAVA 2 “ TMH

References:

- Balaguruswamy “Programming in JAVA”
- Comer “Computer Networks & Internet”
- Deitel & Deitel “Java™ How to Program, 6/E”
- Frouzan “Data communications and Networking”
- Gary Cornell “Core Java” The Sun Micro Systems Series

DESIGNING AND SUPPORTING COMPUTER NETWORK

Course Code: IFT2503

Credit Units: 02

Course Objective:

The objective of the course is to give an overview how to design and support computer network.

Course Contents:

Module I: Introducing Network Design Concepts

Discovering Network Design Basics, Investigating Core Layer Design Considerations
Investigating Distribution Layer Consideration, Investigating Access Layer Design Considerations,
Investigating Server Farms and Security, Investigating Wireless Network Considerations, Supporting
WANs and Remote Workers

Module II: Gathering Network Requirements

Introducing Cisco Lifecycle Services, Explaining the Sales Process, Preparing for the Design Process,
Identifying Technical Requirements and Constraints, Identifying Manageability Design
Considerations

Module III: Characterizing the Existing Network

Documenting the Existing Network, Updating the Existing Cisco IOS, Upgrading Existing Hardware,
Performing a Wireless Site Survey, Documenting Network Design Requirements

Module IV: Identifying Application Impacts on Network Design

Characterizing Network Applications, Explaining Common Network Applications
Introducing Quality of Service (QoS), Examining Voice and Video Options, Documenting
Applications and Traffic Flows

Module V: Creating the Network Design

Analyzing the Requirements, Selecting the Appropriate LAN Topology, Designing the WAN and
Remote Worker Support, Designing Wireless Networks, Incorporating Security

Module VI: Using IP Addressing in the Network Design

Creating an Appropriate IP Addressing Design, Creating the IP Address and Naming Scheme,
Describing IPv4 and and IPv6

Module VII: Prototyping the Campus Network

Building a Prototype to Validate a Design, Prototyping the LAN, Prototyping the Server Farm

Module VIII: Prototyping the WAN

Prototyping Remote Connectivity, Prototyping WAN Connectivity, Prototyping Remote Worker
Support

Module IX: Preparing the Proposal

Assembling the Existing Proposal Information, Developing the Implementation Plan
Planning for the Installation, Creating and Presenting the Proposal

Examination Scheme:

Components	CT1	PR.	ATTD.	EE
Weightage (%)	10	15	5	70

Text & References:

Text:

- CCNA-Discovery 4.0, module 4, Cisco Certified Networking Academy

References:

- Data Communication and Computer Network, Forozoun, TMH Publication
- Data Communication and Network, Stallings, PHI
- Computer Network, Tanenbaum, PHI

JAVA PROGRAMMING LAB

Course Code: IFT2504

Credit Units: 01

Course Contents:

1. Create a "Hello, World" program that simply prints out that statement.
2. Write a program that prints three arguments taken from the command line.
3. Write a program that prints values from 1 to 100.
4. Create a class with a default constructor (one that takes no arguments) that prints a message. Create an object of this class.
5. Write Java assignment statements to evaluate the following equations:
 - (i) $\text{Energy} = \text{mass}(\text{acceleration} * \text{height} + (\text{velocity})^2 / 2)$
 - (ii) $\text{Torque} = 2m_1m_2/m_1+m_2 * g$
6. Design and write a Java program to define a class called Rectangle that contains members for representing its length and breadth. Provide members to get and set these attributes.
7. Design a class to represent a bank account. Include the following members:

Data members:
Name of the depositor
Account number
Type of account
Balance amount in the account

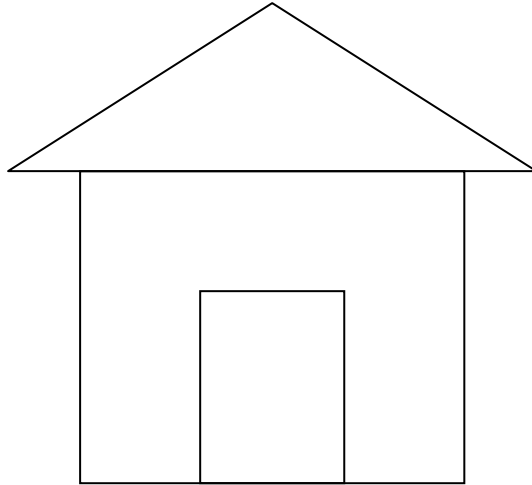
Methods:
To assign initial values
To deposit an amount
To withdraw an amount after checking balance
To display the name and balance
8. Write simple program to calculate the sum of digits of any number.
9. Write a simple program to display a "*" I triangle shape.
Output will be like this

```

*
* * *
* * * * *
```
10. Write a simple program to call a method called simple from a main function. The method simple should accept an integer as an argument and calculate the square of the number in the method simple.
11. Write a Java program to add two integers and two float numbers. When no arguments are supplied, give a default value to calculate the sum. Use method overloading to achieve this.
12. Write a program to perform mathematical operations. Create a class called AddSub with methods to add and subtract. Create another class called MultDiv that extends from AddSub class to use the member data of the superclass. MultDiv should have methods to multiply and divide. A main method should access the method and perform the mathematical operations.
13. Write an interface with a method called display. Implement this method in a class to display two names.
14. Write an interface that has two methods called push and pop of a stack. Write a class to implement the two methods for a fixed size stack creation.
15. Write a small program to catch Negative Array Size Exception. This exception is caused when the array is initialized to negative values.
16. Write a program to handle Null Pointer Exception and use the finally clause to display a message to the user.
17. Write a Java program that takes a string and converts it into uppercase and lowercase letters.
18. Write a Java program to find the volume of a sphere and a cone.
19. Write a Java program to convert rupees to dollars.
20. Write a Java program to find x to the power y. Use overloading for different cases when x and y are combinations of integer and floating point numbers.
21. Create an abstract class called Figure that has an abstract method called draw (). Make the subclasses called Filled_Rectangle, Filled_Arc and override the draw method in which you would print the message regarding the current object.
22. Write a Java program that has integer variables a, b, c and result as float. Store some values in them and apply the formula $\text{result} = a/(b-c)$. Catch the probable exception.
23. Write applets to draw the following shapes:

- (i) Cone
- (ii) Cylinder
- (iii) Cube
- (iv) Square inside a circle
- (v) Circle inside a square

24. Write an applet to display the following figure:



- 25. Write an applet to display five buttons and five labels.
- 26. Write a Java program to create 5 threads by extending Thread class.
- 27. Write a Java program to create 5 threads by implementing Runnable interface.

Examination Scheme:

IA				EE	
A	PR	LR	V	PR	V
5	10	10	5	35	35

Note: IA –Internal Assessment, EE- External Exam, PR- Performance, LR – Lab Record, V – Viva.

DESIGNING AND SUPPORTING COMPUTER NETWORK LAB

Course Code: IFT2505

Credit Units: 01

Course Contents:

1. Process of Core Layer Design Consideration
2. Process of Access Layer Design Consideration
3. Preparing the Design Process, technical requirements.
4. Documentation Design of Existing Network in the Enterprise.
5. Documentation Design of Network Design Requirements.
6. Documenting application and traffic flow in Enterprise network.
7. Selection of the Appropriate LAN Topology
8. Designing WAN and Wireless Network and Incorporating Security.
9. Creating an IP addressing Design with example enterprise network.
10. Prototype Design of LAN, Server Farm and WAN Connectivity.

Examination Scheme:

IA				EE	
A	PR	LR	V	PR	V
5	10	10	5	35	35

Note: IA –Internal Assessment, EE- External Exam, PR- Performance, LR – Lab Record, V – Viva.

BASICS OF OS (UNIX, LINUX AND SHELL PROGRAMMING)

Course Code: IFT2506

Credit Units: 02

Course Objective:

The Objective of this course is to expose the students to the fundamentals and the concepts of Unix Operating System. UNIX commands and Shell Programming will prepare the students to work on UNIX platform, which is the requirement of each industry. This course is designed to understand the concepts of Unix System for Research and Development.

Course Contents:

Module I: Introduction

History of Operating System, Types of Operating Systems, History of Unix, UNIX Family, Unix System Architecture, Kernel, Shell: Types of shells, Files and Directories, Absolute Path and Relative Path, root and hierarchical file structure of Unix, telnet .

Module II: UNIX Commands

Login, password, shell and commands, logout, changing password, current working directory, referring to home directories, Commands to move around by path concept, creating new directories, creating files–touch, cat; copying files; moving files, Deleting files and directories; looking at files: cat, more, pg, less, head, tail; Cal, banner, file, wc, sort, cut, grep ,cmp, comm., diff; Getting online help; manual pages; listing commands, meta characters, Wildcards; hidden files; Standard input and output; redirecting input and output; filter; pipes; file permissions; user and group; Interpreting file permissions; Permission Dependencies; Changing permissions. Managing file links; hard links; symbolic links; jobs and process: process ID; foreground and background jobs; suspend and interrupt a process; killing jobs;

Module III: VI Editor

Command mode, insert mode and last line mode; command to delete character, insert line; deleting text, command for moving the cursor; including other files; running shell commands; getting vi help; search and replace commands; changing and deleting text, Change word, Change line, Delete current line, Delete n lines, Delete remainder of Lines; copying and moving; Saving and Exiting.

Module IV: Shell programming:

Shell variables and Keywords, Shell scripts and execution methods, Interactive execution and command line arguments, (\$1,\$2 etc) meta Characters- Syntactic (&&, (), &, ||, ;:, <, > etc), pattern matching, substitute shell variables, Test Command, Control Flow, For, If, While, Case. String handling and computation using expr, Setting positional parameters (set command), Shift.

Module V: System Administration

Adding and Removing Users, Starting up and Shutting down the System, Disk Management, File System Mounting and Unmounting, Monitoring System Usage, Ensuring System Security

Examination Scheme:

Components	CT1	PR.	ATTD.	EE
Weightage (%)	10	15	5	70

Text & References:

Text:

- Kanetkar. “UNIX Shell Programming” BPB
- Sumitabha Das, “UNIX: Concepts and Application”, TMH.

References:

- Das. “UNIX- Concepts & Applications
- Maurice J. Bach, “Design of the Unix operating System” PHI.

BASICS OF OS (UNIX, LINUX AND SHELL PROGRAMMING) LAB

Course Code: IFT2507

Credit Units: 01

Course Contents:

1. Write a Shell Script that takes a search string and filename from the terminal & displays the results.
2. Write a Shell Script that takes pattern and filename as command line arguments and displays the results appropriately i.e. pattern found/pattern not found.
3. Write a Shell Script that accepts only three arguments from the command line. The first argument is the pattern string, the second argument is the filename in which the pattern is to be searched and the third argument is the filename in which the result is to be stored.
4. Write a Shell Script that accepts a filename as a command line argument and finds out if it's a regular file or a directory. If it's a regular file, then performs various tests to see if it is readable, writeable, executable etc.
5. Write a Shell Script which creates the following menu and prompts for choice from user and runs the chosen command.
 Today's date
 Process of user
 List of files
 Quit to UNIX
6. Write a Shell Script that computes the factorial of a given number.
7. Write a Shell Script that works like a calendar reminding the user of certain things depending on the day of the week.
8. Write a Shell Script that changes the extension of a group of files from txt to doc
9. Write a Shell Script that accepts both filename and a set of patterns as positional parameters to a script.
10. Write a Shell Script which will redirect the output of the date command without the time into a file.
11. Write a Shell Script (using while loop) to execute endlessly (until terminated by user) a loop which displays contents of current directory, disk space status, sleep for 30 seconds and display the users currently logged in on the screen.
12. Write a Shell Script that receives two filenames as arguments. It should check whether content of the two files is same or not. If they are same, second file should be deleted.
13. If a number is input through the keyboard, write a script to calculate sum of its digits.
14. Write a Shell Script that performs a count-down either from 10 (default) or from the value that is entered by the user.
15. Write a Shell Script which takes a command line argument of Kms and by default converts that number into meters. Also provide options to convert km to dm and km to cm.
16. Write a Shell Script using for loop, which displays the message "Welcome to the UNIX System"
17. Write a Shell Script to change the filename of all files in a directory from lower-case to upper-case.
18. Write a Shell Script that examines each file in the current directory. Files whose names end in **old**

are moved to a directory named **old files** and files whose names end in **.c** are moved to directory named **programs**.

19. Write a Shell Script which searches all files in the given directory (to be taken as command line argument) for the file having the title (to be taken as command line argument), as the first line in the file.

- a) Display the contents of the searched file.
- b) In the end, print the file is ###, where
is small-sized if total no. of lines is <50
is medium-sized if total no. of lines between 50&100
is large-sized.

20. Write a shell script which reports names and sizes of all files in a directory (directory would be supplied as an argument to the shell script) whose size is exceeding 1000 bytes. The filenames should be printed in descending order of their sizes. The total number of such files should also be reported.

21. WASS for renaming each file in the directory such that it will have the current shell PID as an extension. The shell script should ensure that the directories do not get renamed.

22. WAP to calculate and print the first *m* Fibonacci numbers.

23. WASS that will receive any number of filenames as arguments. The shell script should check whether such files already exist. If they do, then it should be reported. The files that do not exist should be created in a sub-directory called **mydir**. The shell script should first check whether the sub-directory **mydir** exists in the current directory. If it doesn't exist, then it should be created. If **mydir** already exists, then it should be reported along with the number of files that are currently present in **mydir**.

24. A shell script receives even number of filenames. Suppose four filenames are supplied, then the first file should get copied into second file, the third file should get copied into fourth and so on. If odd number of filenames is supplied then no copying should take place and an error message should be displayed.

25. WASS to identify all zero-byte files in the current directory and delete them. Before proceeding with deletion, the shell script should get a conformation from the user.

26. WASS to compute the **GCD** and **LCM** of two numbers.

27. Two numbers are entered through the keyboard. WAP to find the value of one number raised to the power of another.

28. WASS that prompts the user for the password. The user has maximum of 3 attempts. If the user enters the correct password, the message "Correct Password" is displayed else the message "Wrong Password".

29. WASS that repeatedly asks the user repeatedly for the "Name of the Institution" until the user gives the correct answer.

30. WAP to generate all combinations of 1, 2 and 3 using **for loop**.

Examination Scheme:

IA				EE	
A	PR	LR	V	PR	V
5	10	10	5	35	35

Note: IA –Internal Assessment, EE- External Exam, PR- Performance, LR – Lab Record, V – Viva.

EMERGING TECHNOLOGIES

Course Code: IFT2508

Credit Units: 03

Course Objective:

The course objective is to learn the latest emerging technologies in the Information Technology Field. This course provides the opportunity for students to identify research, gain a basic knowledge of, discuss and evaluate IT-related new and emerging technologies and their impact on information systems, business, and society.

Module I: Intelligent Systems:

Introduction of intelligent systems, rules for intelligent systems, imbibing preposition and predicate logic to Artificial systems, Intelligent reasoning, inference reasoning, Knowledge Acquisition, Learning concept: Machine learning, learning techniques of artificial neural network.

Module 2: Wireless Technologies and Communication System

Wireless Networks, Wireless VS Wired Networks, Mobile Devices, Mobile Applications, Challenges in mobile computing. Cellular architecture, GSM Architecture, General Packet Radio Services (GPRS): GPRS Architecture, GPRS Network Nodes.

Module 3: Decision Support Technologies:

The need for data warehousing, Operational & Informational Data Stores, Data Warehouse definition & Characteristics, Data Warehouse role & Structure, The cost of warehousing data, Data Warehouse Architecture, Data Warehousing Components: Stores, warehouses and marts, Data warehouse database, Sourcing, acquisition, clean up & transformation tools, meta data, Access tools, Data warehouse administration & management.

Introduction to Data Mining & Visualization: Approach of data exploration and data mining, foundation of data visualization.

Module 4: Advanced web technologies

Introduction to WWW architecture, web crawling, web indexing, web searching, J2SE, J2ME, J2EE, concept of .Net Framework: Evolution of .NET, Architecture of .NET framework, Features of .NET, XML: Syntax, XML Schema

Module 5: Computing Paradigms

Introduction, architectures and Characteristics & Disadvantages of Grid Computing, Cluster Computing, Distributed Computing, Cloud Computing

Text & References:

Text :

- Rich E and Knight K, Artificial Intelligence, TMH, New Delhi.
- Kosko: Neural Network & Fuzzy System, PHI Publication
- Alex Berson, Data Warehousing, Data Mining, and Olap, Tata McGraw Hill.
- Cloud Computing: Principles and Paradigms, Editors: Rajkumar Buyya, James Broberg, Andrzej M. Goscinski, Wile, 2011
- XML Complete Reference
- Mobile and Personal Communication systems and services”, by Raj Pandya, Prentice Hall of India, 2001.

References:

- Cloud Computing: Principles, Systems and Applications, Editors: Nikos Antonopoulos, Lee Gillam, Springer, 2012
- S, Rajasekaran & G.A. Vijayalakshmi Pai, Neural Networks, Fuzzy Logic & Genetic Algorithms, Synthesis & Applications, PHI Publication.
- NET programming – Black Book
- C# with Visual Studio – Vijay Mukhi, BPB

SOFTWARE ENGINEERING

Course Code: IFT2509

Credit Units: 03

Course Objective:

The purpose of this course is to acquaint students with the concepts and methods available for software development in industrial environments. Students will be exposed to a variety of topics such as design notations, costing techniques, and testing methods, as well as to the tools which are available to support software specification, design, testing, and maintenance.

Course Contents:

Module I: Software Development Life Cycle

Evolution of Software Engineering, Software Problems, Issues Involved in Software Engineering, Fundamental Qualities of a Software Product, Approaches to Software Engineering, Planning the development Process, Development/Product Life-Cycle Model, Kinds of Software Life-Cycle Model.

Module II: Project Management

Project Management Concepts, Project Management Activities, Size Metrics. Software Requirement analysis and Specification, Cost Models.

Module III: System Design

Design Objectives, Design Principles, Effective Modular Design (Functional Independence, Coupling, and Cohesion), Design Tools and Techniques, Prototyping, Structured Programming.

Module IV: Coding

Programming Practices, Verification, Monitoring and Control.

Module V: Software Testing

Testing Fundamentals, Test case design, Functional Testing, Structural Testing, Test Plan, Activities during testing, Unit System, Integration Testing, Software Maintenance.

Module VI: Software Reliability

Concept of Software Reliability, Software Repair and Availability, Software Errors, Failure and Faults.

Examination Scheme:

Components	CT1	A/C/Q	Attd	EE
Weightage (%)	10	15	5	70

Text & References:

Text:

- Software Engineering, A Practitioner's Approach - Roger S. Pressman.

References:

- ☐ An Integrated Approach to Software Engineering, Pankaj Jalote.
- ☐ Software Engineering Concepts, Richard Fairley.

SUMMER INTERNSHIP – II (EVALUATION)

Course Code: IFT2535

Credit Units: 03

Guidelines:

There are certain phases of every Intern's professional development that cannot be effectively taught in the academic environment. These facets can only be learned through direct, on-the-job experience working with successful professionals and experts in the field. The internship program can best be described as an attempt to institutionalize efforts to bridge the gap between the professional world and the academic institutions. Entire effort in internship is in terms of extending the program of education and evaluation beyond the classroom of a university or institution. The educational process in the internship course seeks out and focuses attention on many latent attributes, which do not surface in the normal classroom situations. These attributes are intellectual ability, professional judgment and decision-making ability, inter-disciplinary approach, skills for data handling, ability in written and oral presentation, sense of responsibility etc.

In order to achieve these objectives, each student will maintain a file (**Internship File**). The Internship File aims to encourage students to keep a personal record of their learning and achievement throughout the Programme. It can be used as the basis for lifelong learning and for job applications. Items can be drawn from activities completed in the course modules and from the workplace to demonstrate learning and personal development.

The File will assess the student's analytical skills and ability to present supportive evidence, whilst demonstrating understanding of their organization, its needs and their own personal contribution to the organization.

The **layout guidelines** for the Project & Seminar Report:

1. File should be in the following specification:

A4 size paper

Font: Arial (10 points) or Times New Roman (12 points)

Line spacing: 1.5

Top & bottom margins: 1 inch/ 2.5 cm

Left & right margins: 1.25 inches/ 3 cm

2. Report Layout: The report should contain the following components:

Front Page

Table of Content

Acknowledgement

Student Certificate

Company Profile (optional)

Introduction

Main Body

References / Bibliography

The File will include **five sections** in the order described below. The content and comprehensiveness of the main body and appendices of the report should include the following:

1. **The Title Page**--Title - An Internship Experience Report For (Your Name), name of internship organization, name of the Supervisor/Guide and his/her designation, date started and completed, and number of credits for which the report is submitted.

2. **Table of Content**--an outline of the contents by topics and subtopics with the page number and location of each section.

3. **Introduction**--short, but should include how and why you obtained the internship experience position and the relationship it has to your professional and career goals.

4. **Main Body**--should include but not be limited to daily tasks performed. Major projects contributed to, dates, hours on task, observations and feelings, meetings attended and their purposes, listing of tools and materials and their suppliers, and photographs if possible of projects, buildings and co-workers.

5. References / Bibliography --This should include papers and books referred to in the body of the report. These should be ordered alphabetically on the author's surname. The titles of journals preferably should not be abbreviated; if they are, abbreviations must comply with an internationally recognised system

ASSESSMENT OF THE INTERNSHIP FILE

The student will be provided with the Student Assessment Record (SAR) to be placed in front of the Internship File. Each item in the SAR is ticked off when it is completed successfully. The faculty will also assess each item as it is completed. The SAR will be signed by the student and by the faculty to indicate that the File is the student's own work. It will also ensure regularity and meeting the deadlines.

STUDENT ASSESSMENT RECORD (SAR)

- **Range of Research Methods used to obtain information**
- **Execution of Research**
- **Data Analysis**
 - Analyse Quantitative/ Qualitative information
 - Control Quality
- **Draw Conclusions**

Examination Scheme:

Components	V	S	R	FP
Weightage (%)	20	20	20	40

V – Viva, S – Synopsis, FP – Final Presentation, R - Report