



INSTITUTE OF HUMAN RESOURCES DEVELOPMENT

TC.86/1949(2), NH Bypass Junction, Chakka, Petta P O, Thiruvananthapuram, Kerala, INDIA-695024

<https://www.ihrd.ac.in>

POST GRADUATE DIPLOMA IN COMPUTER APPLICATIONS

(Two semesters)

Scheme & Syllabus

2024

(Effective from January 2024 admission)

Note: Topics given in italics are included as per G.O. (M.s) No. 10/2019/ITD dt. 04-06-2019, for Elementary and Supervisory level syllabus for standardized IT Courses.



INSTITUTE OF HUMAN RESOURCES DEVELOPMENT

POST GRADUATE DIPLOMA IN COMPUTER APPLICATIONS

(Two semesters)

Subjects of study and Scheme of Assessment

(Scheme-2024)

First Semester

Subject Code	Subject Name	No. of Hrs/Week		Minimum Marks			Maximum Marks		
		T	P	W/P	CE	T	W/P	CE	T
PGDCA101	Operating System	3	-	40	20	75	100	50	150
PGDCA102	DBMS and Database Programming	3	-	40	20	75	100	50	150
PGDCA103	Programming with Python	3	-	40	20	75	100	50	150
PGDCA104	UI/UX Designing & Generative AI Tools for Developers	3	-	40	20	75	100	50	150
GDCA105	C Programming	3	-	40	20	75	100	50	150
PGDCA106	Lab Practice - 1 (Python, Database Interfacing, SQL)	-	3	40	20	75	100	50	150
PGDCA107	Lab Practice - 2 (OS, C and UI/UX Designing)	-	3	40	20	75	100	50	150
Duration: Total 350 Hrs.		15	6	TOTAL MARKS			700	350	1050

Second Semester

Subject Code	Subject Name	No. of Hrs/Week		Minimum Marks			Maximum Marks		
		T	P	W/P	CE	T	W/P	CE	T
PGDCA201	Programming in Java and Software Engineering Concepts	3	-	40	20	75	100	50	150
PGDCA202	Web & Mobile Application Development	4	-	40	20	75	100	50	150
PGDCA203	Developer Tools & Software Testing	3	-	40	20	75	100	50	150
PGDCA204	Computer Networks & Programming	3	-	40	20	75	100	50	150
PGDCA205	Lab Practice - 3 (Java, Software Engineering Concepts)	-	3	40	20	75	100	50	150
PGDCA206	Lab Practice - 4 (PHP, Python, WebApp Frameworks, Android)	-	3	40	20	75	100	50	150
PGDCA207	Project work	-	3	40	20	75	100	50	150
Duration: Total 350 Hrs.		13	9	TOTAL MARKS			700	350	1050

* T - Theory P - Practical W - Written CE - Continuous Evaluation T - Total

Scheme 2024:
PGDCA101: Operating Systems

Module 1: [16 Hrs.]

Introduction to Computer - Familiarity with the basic components of computers and computer terminology, Characteristics of computer, e-governance, multimedia.

Overview of operating systems: Generations, Types, Structure, Services, System Calls, System Boot, System Programs, Protection and Security.

Process management: Process Concepts, Process States, Process Control Block, Scheduling-Criteria, Scheduling Algorithms and their Evaluation, Threads, Threading Issues. Process synchronization: Background, Critical-Section Problem, Peterson's Solution, Synchronization Hardware, Semaphores, Classic Problems of Synchronization, Monitors, Inter Process Communication, Deadlock: System Model, Deadlock Characterization, Deadlock Prevention, Detection and Avoidance, Recovery from Deadlock.

Module 2: [10 Hrs.]

Memory management: Main Memory, Swapping, Contiguous Memory Allocation, Paging, Structure of Page Table, Segmentation, Virtual Memory, Demand Paging, Page Replacement Algorithms, Allocation of Frames, Thrashing.

Module 3: [10 Hrs.]

Storage management: Mass-Storage Structure, Disk Structure, Disk Attachment, Disk Scheduling, RAID Structure. File system interface: File Concept, Access Methods, Directory Structure, File System Structure, Allocation Methods, and Free-Space Management. System Protection: Goals, Principles, Domain of Protection, Access Matrix, Access Control.

Module 4: [14 Hrs.]

Concepts of VIRTUALIZATION – Types, hypervisors, concept of host, guest VMs, Concepts of Data Virtualization, Desktop Virtualization, Server Virtualization, Operating System Virtualization, Network Functions Virtualization. RTOS, Network OS, Cloud Operating Systems – Advantages and Disadvantages – Containers: Docker, Kubernetes

Awareness on LAN, WAN Infrastructure and its components, Installation and configuration of peripherals such as printers, scanners, projectors etc., Awareness on Installation of software, virus scan, Awareness on Data Centre, BCP, DR and its major components

Introduction to GUI Based Operating System, GUI based operating system, File Management, Elements of Word Processing. Awareness on Cyber Security Act and IT Act

Text/Ref:

1. Operating System Concepts, Silberschatz, Ninth Edition, Willey Publication.
2. Operating Systems, Internals and Design Principles, Stallings, Seventh Edition, Pearson Publication.
3. Modern Operating Systems, Tanenbaum, Fourth Edition. Pearson Publication.

Scheme 2024:
PGDCA102: DBMS and Database Programming

Module 1: [15 Hrs]

Introduction to Databases - Overview of Databases - Definition and importance of databases - Evolution of database systems. Data Models - Relational, hierarchical, network, and object-oriented models.

Concept of Relational Database Management System (RDBMS): Basics of Relational Algebra - Introduction to Relational Databases - Entity-Relationship Diagrams (ERD) - Transformation of ERD to relational schema- Tables, rows, columns, Keys (Candidate Key, Primary Key, Foreign Key, Super Key, Alternate Key, Composite Key, Unique Key) – Indexes: Purpose of an Index, How to Create an Index, Types: Primary Index, Clustered Index, Secondary Index - Uses of Primary, Clustered and Secondary Indexes.

Module 2: [15 Hrs]

Introduction to SQL: Definition and concepts of: DDL, DQL, DML, DCL, TCL.

DDL Commands: CREATE, ALTER, DROP, TRUNCATE, RENAME.

Basic SQL commands (SELECT, INSERT, UPDATE, DELETE, MERGE), Aggregate functions, GROUP BY, HAVING. SQL SEQUENCES, Advanced SQL: Subqueries, Joins, Views, Pivot and Unpivot queries, Indexes, Transactions, CALL, Triggers, Stored Procedures

Module 3: [10 Hrs]

Concept of Normalization and Denormalization, Definition and concept of 1NF,2NF,3NF, BCNF, 4NF, 5NF, Pros and cons of normalization.

ORM (Object Relational Mapping), Overview of NoSQL Databases (MongoDB), Types of NoSQL databases (document-oriented, key-value, column-family, graph)

NoSQL MongoDB -Document structure, queries, indexes, Integration with programming languages (Python)

Module 4: [10 Hrs]

Database Security - Access control, encryption, Role-based security

ACID properties - Concurrency control - Emerging Trends and Technologies: Big Data and Databases, Introduction to big data concepts, Hadoop, Spark, and their integration with databases

Cloud Databases: Overview of cloud-based database services (AWS RDS, Azure SQL Database, Google Cloud Spanner)

Introduction to Digital Financial Services- Introduction to Internet based financial services and awareness of various schemes of Govt. of India.

Text/Ref:

1. Modern Database Management 12Th Edition, by Jeff Hoffer, Ramesh Venkataraman
2. Database Systems: The Complete Book by Hector Garcia-Molina, Jeffrey D. Ullman, and Jennifer Widom
3. Learning SQL: Generate, Manipulate, and Retrieve Data 3rd Edition by Alan Beaulieu
4. MongoDB and Python by Niall O'Higgins
5. Cloud Database Development and Management by Lee Chao

Scheme 2024:
PGDCA103: Programming with Python

Module 1: [8 Hrs]

Basic model of computation, Notion of Algorithms, Pseudo code, Flow chart. Programming paradigms – Structured, Functional, Object Oriented, Structured Program Theorem: Sequence, Selection, Iteration, Subroutines, Blocks. Definition of Syntax and Semantics.

Program Development Life Cycle (PDLC) - analysing, designing, coding, debugging, and testing, and documenting, implementing, and maintaining. Concept of Bugs – Classification, Errors, Exceptions. Evolution of Programming Languages, Compilers, and Interpreters. Cross compiling, Concept of TDD.

Module 2: [20 Hrs]

Description of various python distributions (CPython, Anaconda, ActivePython, Jython) – Installations, Features of Python, History and Versions, Applications of Python, Working in Console and Python Shell, Basic Syntax, Variables, Keywords and Comments, Numbers and Strings, Booleans, Operators, converting data types, Control statements: if, if-else, nested if, for loop, while loop, break and continue, pass, match – case, try – except. Generic Data types, list, tuple, dict, set, list comprehension, slicing, sorting of list, dict and tuple, sorted(), reversed().

Input/output functions – formatted strings – string methods, Modules and user defined functions, Iterators, Generators, magic/dunder methods, creating user defined modules, doc string. Functions from Built-in Modules, built-in functions, mathematical functions, date time functions, random numbers, writing user-defined functions, the composition of functions, parameters, and arguments, default parameters, function calls, return statements, using global variables, recursion lambda function. Constructing Modules and Packages.

Module 3: [15 Hrs]

User defined functions – default arguments, key word arguments, variable length arguments, Decorators, Exception handling – Errors, handling multiple exceptions, custom exceptions.

File handling – open, write, read, close, seek, tell. Using context manager “with”. OOPs concepts in python -creating classes, instance methods, inheritance, polymorphism. “SOLID” principles of programming – Single Responsibility Principle, Open-Closed Principle, Liskov Substitution Principle, Interface Segregation Principle, Dependency Inversion Principle.

Connecting to Databases: SQLite – Create, Read, Update, Delete operations using python. Tkinter basics for creating simple GUI in python. Image processing using PIL and Pillow library

Module 4: [7 Hrs]

Concept of Python Enhancement Proposals (PEP), pip command for installing third party libraries, module testing using doctest, unittest. Collections in python: dataclass, namedtuple, deque, ChainMap, Counter, OrderedDict, defaultdict, UserDict, UserList, UserString. Filter, Map, Reduce and Frozenset.

Using ctypes library for loading and executing external dynamic libraries (DLLs, .so) – extending python using C.

Classes and threads – multithreading in python – synchronization.

Text/Ref:

1. Introduction to Computing and Problem-Solving Using Python, E. Balaguruswamy
2. Learning To Program With Python Richard L Halterman
3. Martin C. Brown, Python: The Complete Reference.
4. https://onlinecourses.nptel.ac.in/noc23_cs20/preview

Scheme 2024:
PGDCA104: UI/UX Designing & Generative AI Tools for Developers

Module 1. HTML, CSS [15 Hrs]

HTML Introduction – HTML5 – Structure of HTML Document - Most used tags - Semantic structure tags in HTML5 – Working with Text, Lists, Tables and Frames, Hyperlinks, Images and Multimedia, Forms and controls.

Concept of CSS - Creating Style Sheet (external, internal, inline), CSS Properties, CSS Styling (Background, Text Format, Controlling Fonts), Working with block elements and objects, Working with Lists and Tables, CSS Id and Class, Box Model (Introduction, Border properties, Padding Properties, Margin properties), CSS Advanced (Grouping, Dimension, Display, Positioning, Floating, Align, Pseudo class, Navigation Bar, Image Sprites, Attribute selector), CSS Color, CSS Animations, Creating page Layout and Site Designs.

Introduction Bootstrap (templates and UI components) Buttons, Button Groups, Typography Layout, Responsive images, image carousels Components (Badges, alerts, Breadcrumb), Forms.

Module 2: JAVASCRIPT [15 Hours]

Introduction, variables, data types Array, object, Function Array methods (Advanced) Events Comparisons, Arithmetic operators, date, math If -Else, nested if-Else, switch, Loops- for, while, do-while. Map, Local storage. Validation using JavaScript, Normal function to arrow function.

Asynchronous API in JavaScript, JSON, AJAX (GET, POST), Callbacks, Promises, and async/await, Handling asynchronous code in browsers. Event Loop and Concurrency model - DOM traversal and manipulation techniques - Event handling and delegation - Optimizing performance in DOM operations - Implementing client-side routing with libraries like React Router - Building SPAs with frameworks like React or Vue.js

Module 3. Visual Design and Branding [15 Hrs].

Overview of UI/UX Design - Importance of good design - Basic design principles - User-centered design approach - Color theory and application - shade, tone, temperature, hue, tint. Colour psychology- Typography in UI design - Layout and composition - Iconography and imagery. User research and personas - Information architecture - Wireframing and prototyping - Usability testing. Introduction to interaction design - Animation and transitions – Micro interactions - Designing for various devices and platforms.

Introduction to branding - Components of a brand (logo, tagline, etc.) - Brand identity vs. brand image - Branding strategies and positioning - Principles of effective logo design - Creating a memorable and versatile logo - Case studies of successful logos - Logo design trends.

Module 4: [5 Hrs] . AI Tools for UI/UX Developers (Visual Demonstration of Tools Required, most recent and updated tools currently available in the domain may be brought into the attention of students)

Understanding generative AI - Applications of generative AI in design - Introduction to generative design tools: Uizard, Designs AI, Midjourney, Colormind, Freepik AI image generator, Dall-E, ChatGPT, Khroma, Mockplus, Logolab.

FOSS Tools for UI/UX Design: UI Prototyping tool - Pencil Project, Image Processing - GIMP (GNU Image Manipulation Program), Desktop Publishing – Scribus, Vector Graphics – Inkscape, Image AI Tool - DreamStudio, Canva, Mokker. UI Designer AI - Galileo AI.

Text/Ref:

1. Universal Principles of Design By William Lidwell, Jill Butler, and Kritina Holden
2. Smashing UX Design By Jesmond J. Allen and James J. Chudley
3. Web Design with HTML, CSS, JavaScript and jQuery Set by Jon Duckett (Author)
4. About Face: The Essentials of Interaction Design By Alan Cooper, Robert Reimann, David Cronin, and Christopher Noessel
5. <https://www.oodlestechnologies.com/blogs/10-most-important-and-useful-html5-tags-you-must-know/>
6. <https://pencil.evolus.vn/>

Scheme 2024:
PGDCA105: C Programming

Module 1: [10 Hrs]

Introduction to C Compilers (gcc, Borland/Turbo c) – Stages of compiling a C program – Structure of a simple C Program – Compiling C to executable – debugging – running a C program. ANSI Standard.

Keywords, Identifiers, Constants and Variables - Data types in C - Operators in C - Basic Input and Output Operations - Expressions and Precedence of Operators – standard library functions.

Module 2: [15 Hrs]

Introduction to Control Structures - Branching and looping structures - If statement, If-else statement, Nested if-else, else-if Ladder - switch statement - for loop, while loop, do... while loop, break and continue. Introduction to functions - Function prototype, function definition, accessing a function and parameter passing, recursion – concept of dynamic programming for optimizing recursion - command line arguments – accessing and processing. Header files and macros - preprocessing.

Introduction to Arrays - Declaration and initialization of one dimensional and two-dimensional arrays. Definition and initialization of String - String functions. Concept of Enumeration, Structure, Bit Fields and Union, Declaration and Initialization of structure and union, Nested structures, Array of Structures, Passing structure to functions.

Module 3: [15 Hrs]

Fundamentals of pointers - Declaration, initialization and de referencing of pointers - Operations on Pointers - Concept of dynamic memory allocation - malloc(), calloc(), realloc(), free(), pointers to functions.

Files – Types of file processing: Sequential access, Random access – Sequential access file – Random access file - fopen, fclose, fread, fwrite, fseek, ftell. Error handling during file operations. Working with binary and text files (example code).

Introduction to data structures: what are data structures – importance – concepts of Array list, Linked lists, Stacks, Queues, Deque Binary Trees, Heaps. Algorithms for: Linear search, Binary search. Sorting: Bubble sort, Selection sort, Insertion sort, Merge sort, Quick sort. Concept of Algorithmic complexity notations: Big-O Notation (O-notation), Omega Notation (Ω -notation), Theta Notation (Θ -notation).

Module 4: [10 Hrs]

Names and basic concepts only of Greedy algorithms, Divide and conquer algorithms, Backtracking, String Matching, Bit Manipulation, Randomized Algorithms, Parallel Algorithms, Machine Learning Algorithms, Cryptography Algorithms, Geometric Algorithms, Network Flow Algorithms. Definition of P, NP, NP-Complete, NP-Hard class problems.

Compiling user defined functions to dynamic, static libraries. Using the libraries in other programs – system calls – spawning external programs from c.

Text/Ref:

1. "C Programming Language" by Brian W. Kernighan and Dennis M. Ritchie
2. Programming in ANSI C, 8TH Edition by E Balagurusamy
3. "Data Structures and Algorithm Analysis in C" by Mark Allen Weiss
4. "Data Structures and Algorithms Made Easy in C" by Narasimha Karumanchi
5. <https://www.tutorialspoint.com/how-to-call-a-c-function-in-python>
6. https://reptate.readthedocs.io/developers/python_c_interface.html

Scheme 2024:
PGDCA106: Python, Database Interfacing, SQL
(50 Hrs)

This lab complements the theoretical concepts covered in the Python, Database Interfacing, and SQL course. Students will gain hands-on experience with Python programming, connecting to databases, and utilizing SQL for data manipulation and retrieval.

1. Installing and setting up Python environment (Python/Anaconda/Jython, Jupyter Notebooks) for development
2. Exercises on Basic Python syntax and data types, slicing, string operations
3. Exercises on Control structures (if, loops, match)
4. Exercises on Functions and modules
5. Exercises on File handling in Python
6. Exercises on Exception handling
7. Exercises on Object Oriented Python giving insight into class construction, inheritance, static and class methods
8. Exercises on Creating modules and importing
9. Exercises on Packaging python programs as executable using Pyinstaller
10. Exercises on Tkinter based simple GUI creation
11. Exercises on Turtle Graphics
12. Exercises on Connecting and CRUD operations to SQLite, MySQL, PostgreSQL
13. Exercises on Object Relational Mapping (ORM) – SQLAlchemy
14. Familiarisation of data science libraries like NumPy, pandas, openpyxl, matplotlib
15. Exercises on Generating PDFs using pdfkit
16. Exercises on Python Imaging Library (PIL, Pillow)
17. Create a simple python application based on concepts learned from above exercises.

SQL

18. Exercises on Database Connection to SQLite – CRUD, MySQL
19. Exercises on Creating Database, Tables using SQL
20. Exercises on INSERT, SELECT, UPDATE, DELETE, TRUNCATE, DROP SQL Commands
21. Exercises on Subqueries, Join, Group By, Having, UNION, PIVOT, VIEWS
22. Exercises on setting Primary Key, Foreign Key, Check Constraints
23. Exercises on Transactions, Trigger and Stored Procedures
24. Exercises on User creation and Granting permissions in DBMS.
25. Exercises on Exporting and Importing data from and to DBMS.
26. Exercises on Using ctypes library to load and use external dynamic link libraries

Scheme 2024:
PGDCA107: Lab Practice -2 (OS, C and UI/UX Designing)
(50 Hrs)

This lab accompanies the UI/UX Designing & Generative AI Tools for Developers, C Programming and OS courses, providing hands-on experience with HTML, CSS, JavaScript programming skills and AI Tools for developers. Students will engage in practical exercises to reinforce theoretical knowledge and develop problem-solving skills.

Use of AI tools specified in the theory paper should be encouraged in the lab sessions as far as possible.

UI/UX Development

1. Familiarisation of Structure of HTML documents, HTML elements and attributes, HTML forms
2. Exercise on CSS Selectors, properties, and values, Box model and positioning, Responsive design basics
3. Exercise on CSS Flexbox, CSS Grid
4. Exercise on JavaScript Variables, data types, and operators
5. Exercise on Control flow and loops
6. Exercise on Functions and scope
7. Exercise on Object Model (DOM): Manipulating HTML and CSS with JavaScript
8. Exercise on Events and event handling
9. Exercise on Asynchronous JavaScript Callbacks, Promises, and async/await
10. Exercise on JSON and AJAX
11. Setting up frontend frameworks (React/Angular/Vue.js) development environment
12. Building a simple project with a chosen framework
13. Creating a single-page application
14. Exercise on Customising Website templates
15. Using Design AI create a landing page for e-commerce site.
16. Create a simple website for an educational institution with proper menus, images and page structures
17. Familiarise Generative AI tools for UI/UX Designs – Uizard, Galileo AI, Adobe Sensei, Vance AI, ChatGPT, Invision, Let's Enhance, Khroma etc.

C Programming

18. Exercise on Basic input/output and control structures in C
19. Exercise on File manipulations in C
20. Exercise on Sorting and Searching in C
21. Exercise on Making a dynamic library in C to link with python

OS Commands & IT Department Specifications

1. Linux Terminal Commands (ls, pwd, mkdir, cd, rmdir, cat, cp, mv, rm, uname, locate, touch, ln, clear, ps, man, grep, sort, who, whoami, uniq, paste, cut, echo, cal, wc, sudo, chmod, chown, chgrp, passwd, adduser, deluser)
2. Linux bash shell scripting basics
3. *Office Packages, Elements of Word Processing, Document creation, saving, editing, insertion of tables etc. in document-Create, save, and effectively work with Spreadsheets including formulae and chart, Cell manipulation, Insertion and deletion of rows, columns, chart, graphs, function etc., Demonstrate Presentation Software like Power Point*
4. *Communication and Collaboration - Communication and collaboration tools like Skype, Google docs/sheets etc.- Emails, messaging, Use of Calendars and meeting management using Calendars - Communication and collaboration tools like Skype, Google docs/sheets etc.*
5. *Introduction to Digital Financial Services
Awareness of various Digital Financial services modes (like net banking, mobile banking etc.)
Practical/Hands on sessions:*
6. *Transcription and typing of dictated passage in English and Malayalam
Drafting/Noting using computer
Installation and configuration of peripherals such as printers, scanners, projectors etc.
Create presentations - Create a spreadsheet with data summarization and chart.
Format documents.*

Scheme 2024:
PGDCA201: Programming in Java & Software Engineering Concepts

Module 1: [15 Hrs]

Object-oriented programming concepts – Introduction – Programming paradigms – Advantages and Disadvantages of OOPs - Encapsulation, Inheritance, Polymorphism, Abstraction, Object, Class, Aggregation, Association, Composition, Message passing.

Introduction, Java features, Java Virtual Machine, Java Program Structure, Command Line Arguments. Defining a Class, Creating Objects, accessing class members, Constructors, Garbage Collection, finalize () method, Static Members, Final Variables and Methods, Final Classes, Abstract Methods and Classes, String Handling, Inheritance, Exceptions Handling. Built-in Packages (java.awt, java.io, java.lang, java.math, java.sql, java.util), Creating Interfaces: Defining Interfaces, Extending Interfaces, Implementing Interfaces, Accessing Interface Variables. Creating User Defined Packages, accessing and using a Package.

Module 2: [20 Hrs]

Exception handling-Benefits of exception handling, the classification of exceptions - exception hierarchy, checked exceptions and unchecked exceptions, usage of try, catch, throw, throws and finally, creating own exception subclasses. Multithreading – Differences between multiple processes and multiple threads, thread life cycle, creating threads, interrupting threads, thread priorities, synchronizing threads, inter-thread communication.

Collection Framework in Java – Introduction to java collections, Overview of java collection framework, commonly used collection classes- Array List, Vector, Hash table, Stack, Lambda Expressions. Files- Streams- Byte streams, Character streams, Text input/output, Binary input/output, File management using File class.

Module 3: [10 Hrs]

Java basic GUI programming using swing – Creating a basic Swing application – adding layout, components, form controls: buttons, textbox etc – Listeners and Event processing. Connecting to Database – JDBC Type 1 to 4 drivers, connecting to a database, querying a database and processing the results, updating data with JDBC, Data Access Object (DAO).

Introduction to Java Web Applications: Introduction to JSP, Servlets. Servlet Life cycle - Servlet API & Deployment in Tomcat.

Module 4: [5 Hrs]

Unit Testing using JUnit – example case studies.

Software Engineering: - Definition, importance and characteristics, Principles of Software Engineering: KISS (Keep It Simple, Stupid), DRY (Don't Repeat Yourself), YAGNI (You Aren't Gonna Need It), BDUF (Big Design Upfront), Occam's Razor, Law of Demeter, Avoid Premature Optimization, Measure Twice and Cut Once, Principle of Least Astonishment

Text/Ref:

1. Java: A Beginner's Guide by Herbert Schildt
2. Core Java Volume I: Fundamentals by Cay S. Horstmann
3. JAVA 2 Swing, Servlets, JDBC and JavaBeans Programming Black Book by Steven Holzner (Author)
4. Modern Software Engineering by David Farley
5. Clean Code by Uncle Bob Martin.
6. Head First Design Patterns by Eric Freeman.

Scheme 2024:
PGDCA202: Web & Mobile Application Development

Module 1: [12 Hrs]

Internet - Introduction to Internet, WWW and Web browsers, searching content etc. - Cookies, caches, history etc. Communication and Collaboration - Communication and collaboration tools like Skype, Google docs/sheets etc.- Emails, messaging, Use of Calendars, and meeting management using Calendars – Communication and collaboration tools like Skype, Google docs/sheets etc

Introduction to the Website Development: Concept of Client-Server Model, Web Server, Website and Web Pages, HTTP Protocol, Server-side scripting languages. Scripting Languages Platforms for website development: LAMP, WAMP, MAMP, XAMPP

Introduction to PHP: Basic Syntax, Defining variable and constant, PHP Data types, Operator and Expression. Control structures in PHP, user defined functions, call by value and call by reference, String Creating and accessing, String Searching & Replacing String, Formatting String, Other String Related Library functions. Array: Creating index based and Associative array Accessing array, Element Looping with Index based array, Looping with associative array using each () and foreach(), basic library functions.

Handling Html Form with PHP: Capturing Form, Data Dealing with Multi-value field, and Generating File uploaded form, redirecting a form after submission.

Module 2: [13 Hrs]

Understanding file& directory, Opening, and closing, a file, Copying, renaming, and deleting a file, working with directories, Creating, and deleting folder, File Uploading & Downloading. Introduction to Session Control, Session Functionality What is a Cookie, Setting Cookies with PHP. Using Cookies with Sessions, Deleting Cookies, Registering Session variables, Destroying the variables and Session. Connection with MySQL Database, performing basic database operation (DML) (Insert, Delete, Update, Select), Setting query parameter, Executing query Join (Cross joins, Inner joins, Outer Joins, Self joins.)

Exception Handling: Understanding Exception and error, try, catch, throw. Error tracking and debugging. Object Oriented features in PHP.

Steps for registering a domain name, buying a hosting server and deploying a PHP project to web host using ftp tool.

Module 3: [13 Hrs]

Django Framework: MVT model, setting up a python virtual environment, Installing Django, creating Django project, creating Django web application, running the development server, configuring settings.py, urls.py for a simple project. Django Architecture, Minimal Django project layout.

Django Templating – layouts, Jinja template engine processing commands – setting up database - Adding an Application Generate the application files, Defining models, Related objects, SQL Migration, App configuration, Accessing models. Setting up the admin user, Running the admin site, Tweaking the admin interface, Changing the admin index page.

Django ORM: creating and using models – model-based forms – QuerySets, Field lookups, Chaining filters. Slicing QuerySets. Related fields, Q and F objects, Django Signals.

Forms: Forms overview, GET and POST, The Form class, Processing the form, Widgets, Validation, Forms in templates

Steps for deploying a Django project.

Module 4: [12 hrs]

The Android Platform, Android SDK, Understanding Anatomy of Android Application, Android Manifest file and its common settings. Android terminologies, Application Context, Activities, Services, Intents, Receiving and Broadcasting Intents, Using Intent Filter, Permissions. User Interface Screen elements, Designing User Interfaces with Layouts, Drawing and Working with Animation. Using Android Data and Storage APIs, managing data using SQLite, Sharing Data between Applications with Content Providers, Using Android Networking APIs, Using Android Web APIs, Using Android Telephony APIs, Deploying Android Application to the World.

Text/Ref:

1. Learning PHP, MySQL & JavaScript, 6th Edition by Robin Nixon
2. Python Web Development with Django (Developer's Library) by Jeff Forcier (Author), Paul Bissex (Author), Wesley J Chun (Author)
3. Secure Web Application Deployment Using Owasp Standards: An Expert Way of Secure Web Application Deployment by Subbulakshmi T. (Author), Praveenkumat H. (Author)
4. Android Programming for Beginners: 3rd Edition by John Horton
5. Fundamentals of Android App Development by Sujit Kumar Mishra

Scheme 2024:
PGDCA203: Developer Tools & Software Testing

Module 1: [10 Hrs] Git & GitHub

Version Control System – Importance – Introduction to Git - Git Repository Structure - Working directory, Staging Area (or index), Local Repository, Central Repository - Working of Git Commands: git clone, git config, git checkout, git add, git commit, git push, git pull, git status, git diff, git log, git reset, git branch, git tag, git remote, git help, git rebase. Concept of Forking, workflow, Hook, Working tree. Creating account in GitHub, GitLab and managing repository.

Module 2: [10 Hrs] Docker & Containers

Introduction to Docker - What is Docker - Benefits of using Docker, Docker vs. Virtual Machines, Docker Architecture, Docker Hub

Docker Images: What is an Image, Creating, Dockerfile, Building, Inspecting, Running. Docker Containers: - What is a Container, Running, Starting & Stopping, Inspecting, Restarting, Removing

Docker Networks: - Docker Bridge Network, Docker Host Network, Docker Overlay Network, Connecting Containers to Networks
Docker Volumes: Concept, Creating, Using, Inspecting, Removing Docker Volumes. Docker Compose: Using Docker Compose,
Docker Registry: Public, Private, Pushing Images to a Registry, Pulling Images from a Registry, Docker Security Best Practices.

Module 3: [15 Hrs]

Concepts of DevOps, DevOps principles, Benefits of DevOps, 7 phases of the DevOps lifecycle, Introduction to Jira Software: - concepts of issue, project, board, and workflow.

FTP Tools & Commands: (FileZilla, ftp, sftp, scp), SSH logins using (Putty, ssh), Telnet, Remote Desktop Connection (Windows), Remote Desktop sharing using (TeamViewer, AnyDesk).

Familiarisation of IDEs like PyCharm, IntelliJ IDEA, NetBeans, Eclipse, Visual Studio. Text Editors: Visual Studio Code, Sublime Text. Installing extensions for VS Code, AI extension for code editors.

Basic introduction to “LaTeX” for document generation: Document structure, basic commands for title, table of contents, use packages, including images and graphics, generation of pdf.

Module 4: [15 Hrs]

Introduction: Software Testing, Importance of testing, Roles and Responsibilities, Testing Principles, Attributes of Good Test, V-Model, Test Case Generation, SDLC Vs STLC.

Software Testing Life Cycle: Requirements Analysis/Design, Traceability Matrix, Test Planning, Objective, Scope of Testing, Schedule, Approach, Roles & Responsibilities, Assumptions, Risks & Mitigations, Entry & Exit Criteria, Test Automation, Deliverables.

Types of Testing: Testing Strategies: Unit Testing, Intent Testing, System Testing, Smoke, Regression Testing, Acceptance Testing. Clean Room Software Engineering. Functional/Non-Functional Testing. Testing Tools, Categorization of testing methods: Manual Testing, Automation Testing and Automated Testing Vs. Manual Testing. Notions of Non-Functional Testing, Overview of Test Cases Design, Test Execution, Types of Bugs, Debugging Approaches

Text/Ref:

1. Software engineering- A practitioner's Approach, by Roger S. Pressman, McGraw-Hill International Editions
2. LaTeX: A document preparation system, User's guide and reference manual by Leslie Lamport
3. Software Testing: Principles and Practices by Srinivasan Desikan
4. Docker in Action, Second Edition by Jeff Nicoloff
5. Software Quality Approaches: Testing, Verification, and Validation: Software Best Practice by Michael Haug and Eric W Olsen

Scheme 2024:
PGDCA204: Computer Networks & Programming

Module 1: [10 Hrs]

Types of Networks, Topology, Review of ISO OSI Reference Model and TCP/IP Architecture, Protocols, Ports, Networking devices (Routers, Switches, Hub, Repeater, NIC Cards, Bridge), Networking Transmission Media (Wire, Wireless, UTP, STP, Fiber Optic, Coaxial), The IP protocol, IP Addresses (Version, Classes, Types), Subnets, Subnetting (VLSM, FLSM, CIDR, Super netting), Internet control protocols, Gateway routing protocols, Multicasting.

Module 2: [10 Hrs]

Internetworking: How networks differ, concatenated virtual circuits, Connectionless internetworking, Tunneling, Internetwork routing, Fragmentation, and Firewalls.

Internet Transport Protocols (TCP and UDP): The TCP service model, TCP protocol, TCP Segment Header, TCP connection management, Transmission policy: Congestion control, Timer management, and UDP. Domain Name System: DNS namespace, Resource records, and Name services. SMTP and MIME, HTTP, SNMP, Telnet, FTP, Multimedia.

Module 3: [10 Hrs]

Socket programming: Socket address, Elementary socket system calls, Advanced socket system calls, Reserved ports, Socket options, Asynchronous I/O, Input/Output Multiplexing, Out-of-Band data, Sockets and Signals, Internet Super Server.

Internet architecture, application programming interface (API), Network addressing, Standard ports, UNIX Networking Commands; netstat, ifconfig, ping, dig, nslookup, arp, traceroute, tcpdump, sock, telnet, rlogin, ssh etc, client server concepts.

Module 4:[20 Hrs]

TCP Socket call and UDP Socket call block diagram, Python Socket Module; socket, bind, listen, accept, connect, read, write, close, Basic example: TCP echo server and TCP echo client, UDP echo server, UDP echo client, Python Chat server and Chat client, handling multiple clients at once; the select module, python Threading module.

Python and the web, CGI, twisted (networking framework for Python), some popular python modules: smtplib, httplib, poplib, Programming for the Web; Retrieving web pages with http, Parsing HTML data, XML and XMLRPC, Electronic mail; sending mail, Developing Network server program.

Text/Ref:

1. Foundations of Python Network Programming By: JOHN GOERZEN, APPRESS Publication
2. Andrew S Tanenbaum, David Wetherall, Computer Networks, 5th Edition, Pearson, 2012
3. W. Richard Stevens, Unix Network Programming, Prentice Hall / Pearson Education, 2009
4. James F. Kurose and Keith W. Ross, Computer Networking: A Top-Down Approach Featuring the Internet, 5th Edition, Addison-Wesley, 2012

Scheme 2024:
PGDCA205 Lab Practice – 3 (Java, Software Engineering Concepts)
(50 Hrs)

This lab accompanies the Java Programming & course, providing hands-on experience with operating system concepts and Java programming skills. Students will engage in practical exercises to reinforce theoretical knowledge and develop problem-solving skills.

JAVA

1. Installing and setting up of JDK (Open JDK/ Sun JDK) for development
2. Installation and familiarisation of IDEs and Editors (NetBeans / Eclipse/ VS Code)
3. Exercise illustrating structure of a Java program
4. Exercise on Java syntax and basic data types
5. Exercise on Control Structures in Java
6. Exercise on Classes and objects
7. Exercise on Inheritance and polymorphism
8. Exercise on Exception handling in Java
9. Exercise on Files (Creating, Reading, Deleting, Closing)
10. Exercise on GUI development with Swing or JavaFX
11. Exercise on Event handling in Java
12. Exercise on Multithreading in Java
13. Exercise on Socket programming in Java
14. Exercise on Web development basics with Java (JSP, Servlet)
15. Exercise on JDBC Connecting to databases and CRUD operations
16. Exercise on Writing a simple Test Case using JUnit
17. Creating “jar” files as distributable.
18. Creating a simple application combining required features of Java (GUI or Command Line or WebApp)

SOFTWARE ENGINEERING

19. Familiarisation of Jira, Trello, Basecamp
20. Familiarisation of yED (Graph Editor) – Diagram Editor
21. Familiarisation of Selenium
22. Familiarisation of Postmaster for testing Web APIs

Scheme 2024:
PGDCA206 Lab Practice – 4 (PHP, Python, WebApp Frameworks, Android)
(50 Hrs)

This lab accompanies the Web & Mobile Application Development course, providing hands-on experience with PHP, Python, Frameworks and Android programming skills. Students will engage in practical exercises to reinforce theoretical knowledge and develop problem-solving skills.

1. Setting up a PHP development environment
2. Exercise on Basic syntax, variables, and data types in PHP
3. Exercise on Control structures and loops in PHP
4. Exercise on Connecting PHP to a database and CRUD operations (Create, Read, Update, Delete) in a table
5. Exercise on Python Data types, variables, and operators
6. Exercise on Control Structures in Python
7. Exercise on Functions and modules in Python
8. Setting up Django framework project in Virtualenv
9. Exercise on Routing, templates, and views
10. Exercise on Database integration with Django ORM
11. Exercise on User authentication and authorization
12. Exercise on Middleware and decorators in Django (Setting up debug-toolbar in Django)
13. Exercise on Admin interface in Django
14. Setting up Android Studio,
15. Exercise on Building user interfaces with XML, activities, intents, and navigation, User input and event handling
16. Exercise on Data storage and persistence on Android
17. Create a simple application in Android or Django for displaying clock

Scheme 2024:
PGDCA207: Project Work
(50 Hrs)

The Post Graduate Diploma in Computer Applications (PGDCA) project can cover a wide range of topics depending on the focus and requirements of the program. Below are general directions and steps to guide you in completing a PGDCA project:

1. Project Topic Selection:

- Choose a topic that aligns with your interests and the goals of the PGDCA program.
- Consider areas such as software development, database management, networking, cybersecurity, data science, etc.

2. Project Proposal:

- Create a detailed project proposal outlining the problem statement, objectives, scope, methodology, tools, and expected outcomes.
- Clearly define the technologies or programming languages you will be using.

3. Literature Review:

- Conduct a thorough literature review to understand existing work related to your project.
- Identify gaps in the current knowledge or areas where your project can contribute.

4. System Design:

- Define the architecture and design of your system.
- Create flowcharts, diagrams, and wireframes to illustrate the structure and user interface (if applicable).

5. Implementation:

- Start the development phase based on the design specifications.
- Code your project using appropriate programming languages and tools.
- Regularly document your code for better understanding and future reference.

6. Testing:

- Conduct thorough testing to identify and fix bugs and errors.
- Perform unit testing, integration testing, and system testing.
- Ensure your project meets the specified requirements.

7. Documentation:

- Prepare comprehensive documentation including:
 - Project report with an introduction, literature review, methodology, implementation details, results, and conclusion.
 - **Report typography directions, title page design, certificate and contents page format will be available for download from IHRD home page downloads link. (<https://ihrd.ac.in>)**
 - User manual, if applicable.
 - Source code documentation.
 - Any additional documentation required by the program.

8. Presentation:

- Create a compelling presentation summarizing your project.
- Clearly communicate the problem, methodology, results, and significance of your work.

9. Submission:

- Submit the final project report, documentation, and any other required deliverables by the specified deadline.

10. Evaluation:

- Be prepared for a project presentation or viva-voce where you may be required to explain and defend your work.

11. Project IPR & Utilisation: The intellectual property rights in all project work done by the students shall vest with IHRD, 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 IHRD. 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.

References

1. Project Management for the Unofficial Project Manager by Kory Kogon, Suzette Blakemore, and James Wood
2. Effective project planning and management by W Alan Randolph, Barry Z. Posner
3. Project Management Absolute Beginner's Guide by Greg Horine

POST GRADUATE DIPLOMA IN COMPUTER APPLICATIONS**1. Question paper pattern**

Duration of Exam.: 3 Hrs.

Maximum marks: 100

Part - A Multiple choice / fill in the blanks type questions

Part - B Short Answer type Questions with answer size up to 1 page per question.

Part - C Descriptive type Questions with answer size up to 2 to 3 pages per question.

Marks Distribution

Part	No. of questions.	Need to be answered	Marks/Question	Total
A	20	20	1	20
B	10	8	5	40
C	6	4	10	40
Total		32	...	100

Guidelines for question paper setters:

1. Each part should cover questions from each module in the syllabus.
2. The level of difficulty shall be as follows
 - i) Easy Questions : 30% -40%
 - ii) Intermediate level to difficult : 30% -40%
 - iii) Difficult questions : 20% -30%
3. The question paper setters must prepare and submit the question papers as per the following guidelines.
 - a. Question paper must be designed and prepared to fit in an A4 size paper with one inch margin on all four sides.
 - b. Prepare the Question in MS-Word/Open Office-Write document format. Use only "Times New Roman" font with size 10. Align text to both left and right margins.
 - c. Please leave 5 cm. free area at the top of the front page of each question paper to place examination details/Question paper header by the examination department.
 - d. Avoid placing 1 or 2 questions in the last part in a fresh page, unless it is absolutely necessary. In such case, try to accommodate above questions in the previous page(s) by adjusting top/bottom margins and line spacing, if possible. This will reduce printing expenses.
 - e. Specify marks for each question/part clearly.
 - f. Clearly specify the number of questions to be answered for each Part.
 - g. Confirm that no questions in part B is repeated in Part C also.
 - h. Avoid repeating questions in Part C from the immediate previous examination.
 - i. Key for evaluation must be prepared and enclosed in a separate cover and should be submitted along with the question paper set. Key for evaluation must specify evaluation guidelines for each part in the question paper, otherwise the key prepared will be treated as incomplete.
 - j. Submit Question paper in Laser print out form only. Hand written and printed in poor quality printers is not acceptable.

POST GRADUATE DIPLOMA IN COMPUTER APPLICATIONS**2. Scheme for Continuous Evaluation.**

1. For Theory Papers : Weightage
- a). Average of minimum Two test papers : 30 %
 b). Average of minimum Two Assignments : 30%
 c). Score for Seminar : 20%
 d). Score for Class Attendance. : 10%
 e). Overall performance in the class. : 10%

2. For Practical Papers : Weightage
- a). Average of minimum Two Lab tests : 30 %
 b). Average of minimum Two Lab Assignments : 30%
 c). Maintenance of Lab record : 20%
 d). Score for Lab Attendance. : 10%
 e). Overall performance in the Lab. : 10%

3. Teachers shall submit Mark list for Continuous Evaluation to the Head of Institution in the following format.

Subject:

Sl no.	Regno.	Name	a. Test	b. Assignment	c. Seminar	d. Attendance	e. Performance	Total

4. Head of Institution/Co-ordinator shall forward Continuous evaluation marks to the Examination Department in the following format only.

Centre code: Centre name:

Sl no.	Regno.	Name	PGDCA101	PGDCA102	PGDCA103	PGDCA104	PGDCA105	PGDCA106	PGDCA107
			50	50	50	50	50	50	50

5. Continuous evaluation (CE) marks must be published in the notice board at least one week before the commencement of theory examinations after getting approval from the Head of Institution/Co-ordinator.

Thiruvananthapuram
16.02.2024

Sd/-
DIRECTOR