1. Concepts of Programming & Operating System — 66 Hours

Getting Started, Variables & Methods, Operators, Conditional and Looping Statements, Objects, Arrays, Introduction to OS, Introduction to Linux, Shell Programming, Processes, Signals, Threads, Memory, management, Virtual Memory, Deadlock, Inter process communication

2. Object Oriented Programming with Java — 104 Hours

Features of Java, JVM Architecture, Primitive data types, Object Oriented Programming Concepts, Interfaces, Arrays, Garbage collection, Wrapper Classes and String Class, Exception Handling, java.io & java.nio Package, Object Class & java.util Package, Collections, MultiThreading, Synchronization, Inner Class, Lambada Expression

3. Algorithms and Data Structures Using Java — 82 Hours

Problem Solving & Computational Thinking, Algorithms & Data, Structures, Basic Data Structures, Linked List Data Structures, Recursion, Trees & Applications, Searching Algorithms, Sorting Algorithms, Hash Functions and Hash Tables, Graph & Applications, Algorithm Designs, Analysis of different type of Algorithms, Data Structure Implementation and Applications

4. Database Technologies — 72 Hours

DBMS, MySQL, Database Design, Entity-Relationship Diagram, Codd's 12 rules for RDBMS, SQL, Categories of SQL Commands, Normalization, MySQL Data Types, Database Constraints, SQL Functions & Operators, Joins, Subquery, Views, Indexes, ACID Properties, Stored Procedures, Cursors, Triggers, Introduction to NoSQL, MongoDB, Introduction to Big Data

5. Web Programming Technologies — 108 Hours

Architecture of Web, HTML, Cascading Style Sheets (CSS), JavaScript, jQuery, JSON & Ajax, Node.js, Node.js Asynchronous Programming, Node.js Modules, Introduction to Express, React, Introduction to React-Redux, Responsive Web Design & Web Security

6. Web-based Java Programming — 100 Hours

J2EE Overview, Servlets, JSP, Sessions, JDBC & Transaction Management, Hibernate Framework, Spring Framework, Spring Boot, Spring Data Module, Spring AOP, Building REST Services with Spring, Testing in Spring, Securing Web Application with Spring Security

7. Microsoft.Net Technologies — 84 Hours

.Net Framework, Visual Studio, C# Basics, Interfaces, Indexers, Generic classes, Collections, Delegates, Lambdas, Error Handling (Exceptions Handling), LINQ to objects, PLINQ, Files I/O and Streams, Threading

Asp.Net MVC, MVC State Management, MVC Module, Data Management with ADO.NET, Understanding Routing & Request Life Cycle, Layouts, Bundle, Minification, MVC Security, Entity Framework, Understanding ASP.Net MVC Core, Windows Communication Foundation, Web APIs

8. Software Development Methodologies — 84 Hours

Git, Software Engineering, Software Development Life Cycle, Object Oriented Analysis and Design, Agile development model, Introduction to Atlassian Jira, Microservices, API gateway, DevOps, Containerization, Docker, YAML, Kubernetes, Software testing, Quality Assurance vs Quality Control vs Testing, STLC and V Model, Manual testing, Automation testing, Selenium , Jenkins,

Introduction to Cloud, Cloud architecture, Service models: IaaS, PaaS, SaaS, Deployment models: Private, Public, Hybrid, Introduction to AWS

9. General Aptitude — 32 Hours

Percentage, Profit & Loss, Ratio & Proportion, Average, Mixture & Alligation, Simple Interest & Compound Interest, Number Systems, Series, Cyclicity & Remainders, Data Interpretation, Syllogism, Coding & Decoding, Blood Relations, Seating Arrangements (Linear & Circular), Ages, Puzzles, Time, Speed & Distance, Trains, Boats & Streams, Time & Work, Wages (Man days), Pipes & Cisterns, Clocks, Permutations & Combinations, Probability, Calendar

10. Effective Communication — 48 Hours

Fundamentals of Communication, The Art of Communication, Personality Development, English Grammar, Correct Usage of English, Common Mistakes in English Communication, Listening Skills, Reading Skills, Writing Skills, Public Speaking, Presentation Skills, Group Discussions, Interpersonal Skills, Personal Interviews

11. Project — 120 Hours

In addition to the specific subject knowledge, the Software Project module attempts to put into practice a number of things that the students have learned during the PG-DAC course, such as:

- Ability to work in a team
- Software development methodology and principles
- Good programming practices
- Technical reporting and presentation.

The Software Project module is divided in three phases.

I – SRS Phase:

Tasks: Requirements gathering, feasibility study and project thinking.

II – Design Phase:

Tasks: Software design and project plan.

III – Development Phase:

Tasks: Coding and testing of the software system/application.