C++ Programming

Getting Started, Operators & Expressions, Conditional and Looping Statements, Functions in C++, Memory Management and Pointers, OOP Concepts using C++, Constructor and Destructor, Inheritance, Polymorphism, Virtual Functions & Abstract Class, Exception Handling, Managing Console I/O Operations, File Handling in C++ & Templates, STL and RTTI

Concepts of Operating System & Software Development Methodologies

Concepts of Operating Systems:
Introduction to OS, Introduction to Linux, Shell programming, Processes, Memory management, Virtual Memory, Deadlock.
Software Development Methodologies:
Git, Software Engineering, Software Development Life Cycle, Object Oriented Analysis and Design, Agile development model, Introduction to Atlassian Jira, DevOps, Containerisation, Docker, YAML, Kubernetes, Software testing, STLC and V Model, Manual & Automation testing, Selenium, Jenkins

Object Oriented Programming with Java

Introduction to Java, JVM Architecture, Primitive data types, OOP Concepts using Java, Interfaces, Arrays, Garbage collection, Inner Class, Wrapper Classes and String Class, Exception Handling, java.io & java.nio Package, Object Class & java.util Package, Collections, MultiThreading, Synchronization, Lambda Expression, Generics and Reflection API, Java 8 Stream API

Algorithms and Data Structures (Using Java)

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

Database Technologies

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

Web Programming Technologies

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

Web-based Java Programming

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

Microsoft .Net Technologies

.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, MVC Integration with React.

General Aptitude & Effective Communication

Aptitude: Percentage, Profit & Loss, Ratio & Proportion, Average, Mixture & Allegation, 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

Effective Communication: 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

Final Project

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.