Note: *This course outline and schedule are tentative and may be adjusted by the instructor depending on class progress and circumstances.*

Software Engineering

Course Outline

Course Details

This Software Engineering program is designed to provide students with a strong foundation in programming, application development, and modern frameworks. The curriculum covers **C Language, C#, MVC .NET, and Python** to ensure learners acquire versatile coding skills for both desktop and web applications. Students may enroll in the **complete program** or select **individual modules** based on their career goals.

Course Type: Diploma
Course Duration: 1 Year
Class Frequency: 3 classes/week, 2 hours each
Mode of Delivery: On-Campus

Course Objectives

By the end of this course, students will be able to:

- Write efficient programs in **C Language** using memory management, data structures, and file handling.
- Develop **desktop and web applications** using **C# and MVC .NET**.
- Build **full-stack web solutions** using **Python and Django**.
- Apply object-oriented programming, API integration, and database management.
- Work on **real-world projects** and build a professional portfolio.

Note: *This course outline and schedule are tentative and may be adjusted by the instructor depending on class progress and circumstances.*

COMPUTER COLLEGI@TE
www.computercollegiate.com.pk

Week-wise Course Plan

Module 1: C Language (2 Months | 8 Weeks)

Week	Topics Covered	Learning Objectives	Assignments
Week 1	Introduction to C, Language Fundamentals	Understand history, structure, and importance of C language; Write first program	Write a "Hello World" program & identify components
Week 2	Variables & Data Types	Learn variable declaration, constants, type casting	Create a program using different data types
Week 3	Operators (Arithmetic, Relational, Logical)	Apply operators in problem-solving	Write a calculator program
Week 4	Control Statements (Loops, if-else, switch)	Implement decision-making and looping logic	Build a program with nested loops
Week 5	Arrays & Strings	Understand single/multi-dimensional arrays, string handling	Write a program to reverse a string
Week 6	Pointers & Structures	Master pointers, memory management, and structures	Create a student database using structures
Week 7	Functions (Built-in & User-defined)	Use modular programming techniques	Build a program with at least 3 user-defined functions
Week 8	File Management + **Mini Project**	Perform file operations (read/write), integrate all concepts	Develop a small project (e.g., Student Management System)

Note: *This course outline and schedule are tentative and may be adjusted by the instructor depending on class progress and circumstances.*

Module 2: C# (3 Months | 12 Weeks)

Week	Topics Covered	Learning Objectives	Assignments
Week 1	Introduction to OOP Concepts	Understand classes, objects, and inheritance	Create a simple class-based program
Week 2	Fundamentals of C#	Write console programs with variables, methods	Create a calculator in C#
Week 3	Data Types & Decision Statements	Use conditions and branching	Build a grade evaluation system
Week 4	Loops & Exception Handling	Handle errors gracefully	Program with try-catch for division
Week 5	C# in Console Applications	Develop interactive apps	Build a console-based quiz
Week 6	Windows Forms Basics	Create GUI-based applications	Design a student record form
Week 7	SQL Server Introduction	Learn database concepts	Create a database with tables
Week 8	Connecting C# with SQL	Perform CRUD operations	Build a simple inventory system
Week 9	ADO.NET	Work with datasets and commands	Display database records in a form

Note: *This course outline and schedule are tentative and may be adjusted by the instructor depending on class progress and circumstances.*

Week	Topics Covered	Learning Objectives	Assignments
Week 10	Crystal Reports	Generate reports from database	Create a student marks report
Week 11	Advanced OOP Features	Apply polymorphism, abstraction	Build an inheritance-based system
Week 12	**Final Project**	Integrate concepts	Develop a desktop management app (Library / Billing System)

Module 3: MVC .NET (3 Months | 12 Weeks)

Week	Topics Covered	Learning Objectives	Assignments
Week 1	Introduction to ASP.NET	Understand ASP.NET environment	Build a simple web app
Week 2	Web Forms & Master Pages	Create reusable layouts	Build a multi-page site
Week 3	MVC Concepts	Understand MVC architecture	Create a simple MVC app
Week 4	Controllers & Views	Work with controllers, Razor syntax	Display data dynamically
Week 5	Models & Data Annotation	Validate user input	Add validation to a form

Note: *This course outline and schedule are tentative and may be adjusted by the instructor depending on class progress and circumstances.*

Week	Topics Covered	Learning Objectives	Assignments
Week 6	Partial Views	Reuse UI components	Build navigation using partial views
Week 7	SQL Integration	Connect MVC with SQL DB	CRUD operations in MVC
Week 8	Entity Framework (Code First)	Use EF ORM	Create DB schema from code
Week 9	Authentication & Authorization	Secure web applications	Implement login/logout system
Week 10	Web API Basics	Develop REST APIs	Create API for a to-do app
Week 11	Advanced MVC Features	Routing, Filters, Bundling	Optimize app performance
Week 12	**Final Project**	Complete web application	Develop a full MVC project (e.g., E-Commerce Website)

Note: *This course outline and schedule are tentative and may be adjusted by the instructor depending on class progress and circumstances.*

Module 4: Python (4 Months | 16 Weeks)

Week	Topics Covered	Learning Objectives	Assignments
Week 1	Python Fundamentals & Syntax	Learn basic syntax, variables	Write a calculator program
Week 2	Data Types & Arrays	Work with lists, tuples, dictionaries	Create a student list system
Week 3	Control Flow (if, loops)	Apply decision-making	Build a number guessing game
Week 4	Functions	Write reusable code	Program with custom functions
Week 5	OOP in Python	Classes, inheritance, polymorphism	Build a class-based system
Week 6	Exception Handling	Error detection and handling	File read with error handling
Week 7	File Handling (JSON, CSV)	Manage data storage	Write a program to store data in CSV
Week 8	Python Modules & Packages	Import and use libraries	Use `math` and `os` modules
Week 9	Django Introduction	Learn MVT structure	Build a basic Django app

Note: *This course outline and schedule are tentative and may be adjusted by the instructor depending on class progress and circumstances.*

Week	Topics Covered	Learning Objectives	Assignments
Week 10	Django Models & Serializers	Connect DB and APIs	Create a blog database
Week 11	Templates & HTML Wrapping	Create dynamic pages	Build a portfolio website
Week 12	URL Routing (urls.py)	Implement navigation	Add routing to blog project
Week 13	Django Forms & Validation	Handle input safely	Build a contact form
Week 14	Django REST Framework	Build RESTful APIs	Create API for student data
Week 15	Advanced Django Features	Middleware, authentication	Add login/signup system
Week 16	**Final Project**	Complete web application	Develop a full Django project (e.g., Online Store)

Note: *This course outline and schedule are tentative and may be adjusted by the instructor depending on class progress and circumstances.*

Assessment Structure

- **Weekly Assignments** – 20%
- **Mid-Module Tests** – 20%
- **Class Participation & Attendance** – 10%
- **Final Project (Each Module)** – 25%
- **Final Comprehensive Project** – 25%

Final Project

Students must complete a **capstone project** at the end of the course (or end of chosen module). Example projects:

- Student Management System (C Language)
- Inventory Management System (C#)
- E-Commerce Website (MVC .NET)
- Online Store / REST API (Python Django)

Recommended Resources

- **Books:**
  - *The C Programming Language* – Kernighan & Ritchie
  - *C# in Depth* – Jon Skeet
  - *Pro ASP.NET MVC 5* – Adam Freeman
  - *Python Crash Course* – Eric Matthes
- **Tools & IDEs:** Visual Studio, PyCharm, SQL Server, GitHub
- **Online Resources:** W3Schools, MDN, Django Documentation, Microsoft Learn

Attendance Policy

Regular attendance is essential for successful course completion. Students are expected to attend at least **80% of classes in each module**. More than **20% unexcused absences** may result in disqualification from the final project and certification.

Note: *This course outline and schedule are tentative and may be adjusted by the instructor depending on class progress and circumstances.*