Programming Course

Introduction

Join our transformative programming course covering C++, Java, and Python. Start with C++ basics, flow charts, and setting up the development environment. Move on to control flow, functions, arrays, and memory management. Discover classes, objects, and constructors.

In intermediate C++, explore templates, STL, and advanced topics like operator overloading. Learn Java's history, features, and syntax. Master OOP, packages, and I/O. Dive into Java collections, generics, and file handling. Explore networking, JDBC, and servlets.

In Python, grasp syntax, data types, and control flow. Work with functions, modules, and files. Handle data structures, OOP, and GUI programming. Get into web development with Django or Flask, databases, and performance optimization.

Collaborate on projects, use version control, and practice best coding practices. Become a versatile programmer equipped to tackle real-world challenges. Unleash your potential in this 6-month journey of learning and growth.

REQUIREMENTS:

- 18 years and above
- Basic knowledge and interest in entrepreneurship, start-ups and marketing
- Familiar with basic IT Skills

CURRICULUM:

Sr. No.	Contents
Month 1 Intermediate C++ Programming	
Week 1	• Introduction flow charts, pseudocodes • Setting up the C++ development environment (IDE, compiler) • Basic syntax and data types in C++
Week 2	• Control flow statements (if-else, loops, switch) • Functions and modular programming in C++ • Basic input/output operations
Week 3	• Arrays and strings in C++ • Pointers and memory management • File handling and exception handling in C++
Week 4	• What are Classes • What is an object • rendering a class (Constructors, Overloaded Constructors, data members)
Month 2 Intermediate Concepts	

Sr. No.	Contents
Week 5	• Templates and generic programming in C++ • Standard Template Library (STL) containers and algorithms • Smart pointers and memory management techniques
Week 6	• Advanced topics in C++: operator overloading, namespaces • Exception handling and error reporting • Debugging and optimization techniques in C++
Month 3 Java Programming	
Week 7	• Introduction to Java: history, features, and applications • Setting up the Java development environment (IDE, JDK) • Java syntax, data types, and control flow statements
Week 8	• Object-oriented programming (OOP) in Java: classes, objects, and inheritance • Packages and access modifiers in Java • Java I/O and exception handling
Week 9	• Java collections framework: ArrayList, LinkedList, HashMap, etc. • Generics and type safety in Java • File handling and serialization in Java

Sr. No.	Contents
Week 10	• Inheritance and its types • Polymorphism
Month 4 Advanced Java Programming	
Week 11	• Networking in Java: sockets, TCP/IP, UDP • Database programming with JDBC • Introduction to Java Servlets and JSP
Week 12	• Java web development with frameworks like Spring or Java EE • Introduction to Hibernate or other ORM frameworks • Java unit testing and debugging tools
Month 5 Python Programming	
Week 13	• Introduction to Python: history, features, and applications • Setting up the Python development environment (IDE, interpreter) • Python syntax, data types, and control flow statements
Week 14	• Functions, modules, and packages in Python • File handling and exception handling in Python • Working with libraries and external modules in Python

Sr. No.	Contents
Week 15	• Data structures in Python: lists, tuples, dictionaries, sets • Object-oriented programming (OOP) in Python • Python standard library and built-in modules
Week 16	• Python GUI programming with Tkinter or PyQt • Web scraping and data manipulation with Python • Introduction to data analysis with NumPy and pandas
Month 6 Advanced Concepts	
Week 17	• Web development with Python: Django or Flask framework • Working with databases using Python • Python debugging and testing techniques
Week 18	• Deploying Python applications • Performance optimization and profiling in Python • Security considerations and best practices in Python
Week 19	• Project development using C++, Java, and Python • Collaborative coding, version control (e.g., Git) • Documentation and code commenting best practices

Outcomes:

- Enhanced Problem-Solving Skills: Upon completing the programming course, students will demonstrate improved problem-solving abilities. They will be proficient in breaking down complex tasks into smaller, manageable steps and employing logical thinking to develop efficient solutions.
- Proficiency in Coding Languages: Participants will gain hands-on experience in multiple programming languages, such as Python, Java, C++, or JavaScript.
- Creation of Real-World Applications: The course will empower students to apply their programming knowledge practically.
- Increased Employability and Career Opportunities: Completing the programming course will significantly enhance participants' employability in the technology industry.

BENEFITS:

- High Demand and Lucrative Career Opportunities: Graduates of programming courses often find themselves with numerous job opportunities and competitive salaries in fields like software development, web development, data science, and more.
- Flexibility and Versatility: Programming knowledge provides a versatile skill set that can be applied to diverse projects and industries
- Problem-Solving and Critical Thinking: Programming courses cultivate strong problem-solving and critical thinking abilities.
- Continuous Learning and Adaptation: The world of technology is constantly evolving, and programming courses teach students how to adapt and stay up-to-date with the latest advancements.

Affiliation & Collaboarations

PBTE


PUNJAB SKILLS DEVELOPMENT AUTHORITY
GOVERNMENT OF THE PUNJAB
PSDA