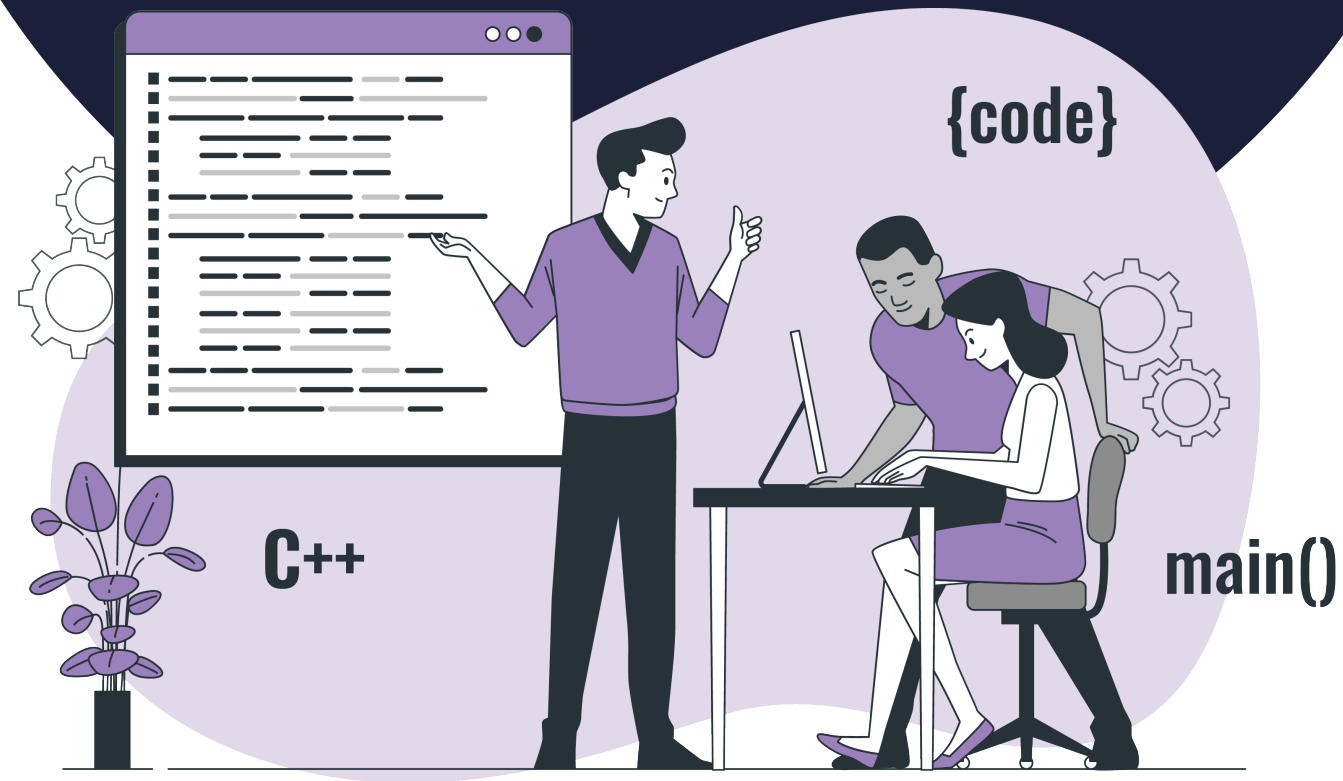
Lesson :

Introduction to Programming and C++



PW SKILLS

List of Concepts Involved:

- What is Programming?
- Why do we need Programming?
- Types of Programming Languages
- Introduction to C++
- Difference between C and C++

Topic 1: What is Programming?

Programming is simply creating a set of instructions for a computer to perform a specific task. Programming can be done using a variety of computer programming languages eg. C,C++,Java and the list is endless !. The best part is, out of these very many options, you can go ahead with the language of your choice to implement the concepts in the form of a logical code.

A Common thought - I have not done programming ever before! How will I be able to learn ?!
There are many instances where we use programming concepts without even having the slightest idea. Surprised ?! Have you ever set the alarm in your watch/phone? I am sure, all of you had during exams !!! When you are setting the time for the alarm, you are actually giving the set of instructions (unknowingly) to the clock/phone to ring (and vibrate) at a specific time. Sounds interesting? Lets see more such examples:

- Many of the coffee lovers here would relate to this instance !
  After you set your coffee machine to start at a certain time in the morning, you're programming an activity to trigger. Most of these straightforward machines utilize programming to execute straightforward tasks— such as setting a timer.
- Travel itineraries or the grocery list that we often create are also programs, wondering how ? They are actually the set of instructions for the supercomputer named 'your brain'. Cool, isn't it?

To know and accomplish more of such interesting things , let's now move to the learning part of programming with programming languages.

Category of Programming Languages

Programming languages that have been developed so far can majorly be categorised into: machine language, assembly language, and high-level language.

Machine Level Language:

As the name suggests, it is a low-level language made up of binary numbers or bits that a machine/computer can understand. It is a sequence of 0s and 1s. It is also known as machine code or object code. Any program that we write, finally transforms into the sequence of 0s and 1s for the computer to understand.

Assembly language:

Assembly language is intended to communicate directly with a computer's hardware. Unlike the usual machine language, which consists of binary characters, assembly language is expressed in a more human readable form. In future many of you are going to learn and program microprocessors in your higher studies or jobs; that would require an expertise in assembly language.

High-level language:

It refers to the programming languages that allow the use of symbolic operators (to signify operations, eg.+,-,*,/ etc.) and symbolic names(to represent data and data structures, eg. variable names).They are also structured with syntax and semantics to describe the computing algorithm/program.This often requires the use of a compiler or an interpreter which helps in the translation from high level human understandable code into low level machine code because the computer only works on the binary data. We will be learning and focussing on this category of language in our course.

Topic 2: Why do we need Programming?

Computers are ridiculously fast but contrary to the assumption of being 'smart', they are extremely dumb machines. You might be wondering, what ?! Can you imagine, they don't even know how to check if a number is odd or even (which a 2nd grade child knows) but if we create a program which will enable them to do this task, they can check a billion numbers for us in one second. Hence, to make a computer/machine 'smart' we need the tool of programming.

Most of the programming is done for the real world applications and not just for big rocket or other scientific missions. Internet banking, online shopping ticket booking, in flight, all this can be conceivable due to the applications created by computer programming. Your washing machine too contains a few sorts of computer programs. All these things and many more are possible through the magic of programming.

Lets now see what are the different approaches which we can follow while programming. These approaches, in computer terminology, are referred to as paradigms. There are many types of paradigms that can be used based on the complexity of our application but we will focus on the most widely used here :

Topic 3: Types of Programming Paradigms :

- Procedural
- Functional
- Object Oriented

Procedural:

Consider a scenario where a program has to collect some data from users (read), perform some calculations (calculate), and then display the result(print) when requested; a simple example of this scenario is any online transaction that you/your parents do. In procedural approach, we can write 3 different procedures/functions for each of the operations of reading, calculating and printing (which could interact amongst each other as well). Hence, in the procedural approach:

- The entire program code is organized as a set of procedures/functions or discrete blocks of codes that are executed in an order.
- These procedures are also known as subroutines or functions and contain a series of steps that will be carried out when the procedure is called/used.
- Some of the programming languages that enable us to use procedural approach : BASIC, FORTRAN, ALGOL, C, COBOL, and Pascal.

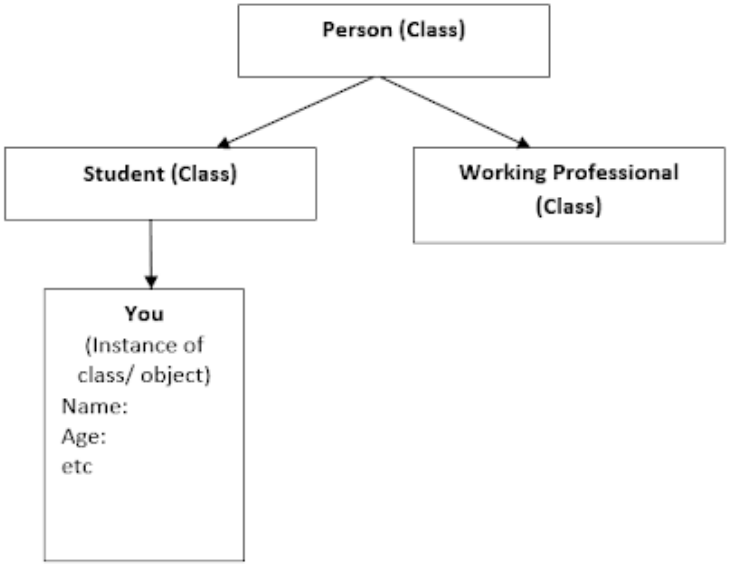
Functional:

Let's assume a scenario where we have to perform lots of different operations on the same set of data. This can be in the domains such as the web, statistics, financial analysis, and machine learning. Here, the functional programming paradigm helps a lot.

- The program code is organized in pure functions(which always yield the same value for the same set of inputs without any deviation) with each function set up to perform a clearly defined task.
- The data/variables are passed in the function as parameters (for the functions to interact with other functions or programs).
- Languages that support this approach are: Javascript, Python, etc.

Object Oriented:

- The program data is organized in classes and objects rather than functions and logic.
- A class is a blueprint for creating objects and an object is referred to as an instance of a class that has unique attributes and behavior.
- A good example of this could be you, yes you! You can be a 'student' or a 'working professional'. In this scenario, under the class of 'Person', we can have a 'Student' or 'Working professional' class and you can be an instance of any of these based on whether you fall in the 'student' category or 'working professional' category. Hence, you are an object



- Languages that support object-oriented approach include: Java, C++, C#, Python, R, PHP, Visual Basic.

The programming language that we would learn in the very first go has to be a simple yet most capable one!
Yes, you guessed it right, it's C++ !!
Lets now, hit the bulls eye !!
Let's get started.

Topic 4: Introduction to C++

- C++ is a popular programming language.
- C++ is a middle-level programming language.
- C++ is developed by Bjarne Stroustrup in 1979 at Bell Labs.
- C++ runs on a variety of platforms, such as Windows, Mac OS, and the various versions of UNIX.
- C++ is a superset of the C language.
- C++ is closer to the compiler and faster than C language.
- C++ is a statically typed programming language i.e it uses static typing when type checking is performed during compile-time as opposed to run-time.
- C++ is both a procedural and an object-oriented programming language. It supports OOP features such as polymorphism, encapsulation, and inheritance.
- As C++ is an object-oriented programming language so it gives a clear structure to programs and allows code to be reused, lowering development costs.

Topic 5: Some basic differences between C and C++ before we move ahead:

- C is a structural programming language that doesn't provide any support for classes and objects but C++ is an object-oriented programming language, and it provides support for the concept of classes and objects.
- C primarily supports procedural programming for developing codes whereas C++ supports both programming paradigms- procedural as well as object-oriented.
- The C language uses the scanf() function for the input operation and printf() function for the output operations whereas C++ language uses the cin function for the input operation and cout function for the output.
- C++ is closer to the compiler and faster than C language.

Upcoming Class Teasers:

- Installation of vs code.