Lesson:

Java

Introduction to Programming

PW SKILLS

List of Concepts Involved:

- What is Programming?
- Why do we need Programming?
- Types of Languages
- Introduction to Java
- Features of Java Programming
- Java architecture
  - JVM
  - JRE
  - JDK
  - Difference between JRE, JDK, JVM

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. Java, C++, C 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/coding ever before! Will I be able to learn it ?!
There are many instances where we use programming concepts without even having the slightest idea. Surprised ?! Have you ever set the alarm on 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. So, yes! You already are a programmer. Sounds interesting? Let's see more such examples:

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 called 'your brain'. Cool, right?

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 categorized 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 a 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 the 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.

Before we move ahead, are you still confused about the differences between compiling and interpreting? Let us discuss this.

Compilation Vs Interpretation:

Compilation is the process to convert a high-level programming language into machine language (all at once) that the computer can understand. The software which performs this conversion is called a compiler.

Interpretation is performed by an Interpreter which directly executes instructions written in a programming or scripting language without previously converting them to an object code or machine code. This conversion of code happens line by line.
Examples of interpreted languages are Perl, Python and Matlab.

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 that 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 rockets or other scientific missions. Internet banking, online shopping, flight ticket booking, 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.

Let's now see what are the different approaches 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 ones.

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.

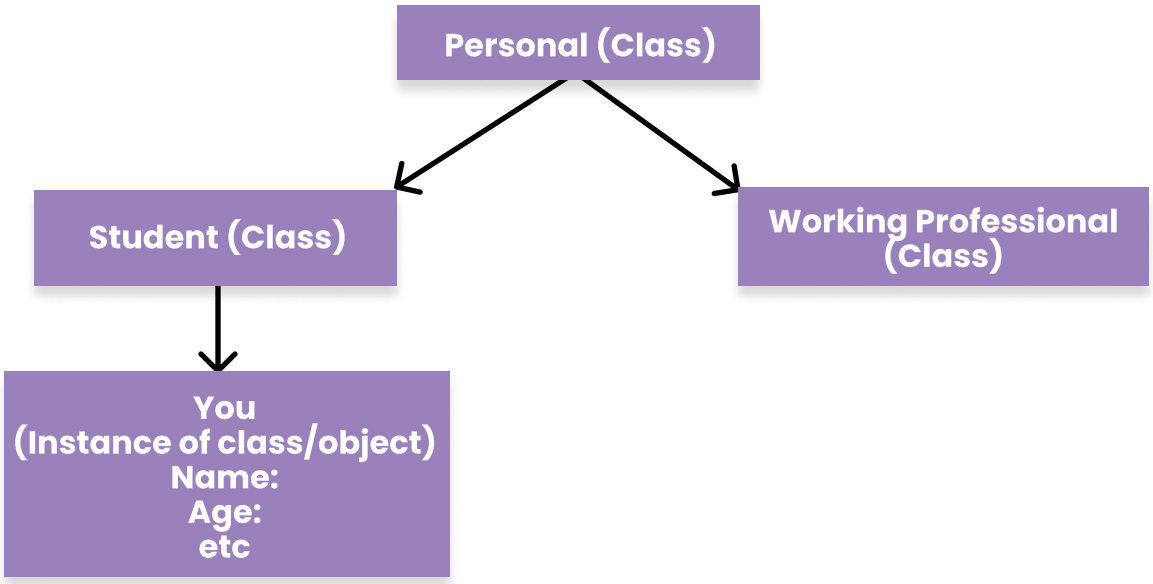
Some of the programming languages that enable us to use procedural approach : BASIC, FORTRAN, ALGOL, C, COBOL, and Pascal.

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 are going to learn here is extremely simple yet very powerful! Any guesses ?! Yes, it's Java (we are already in the course, no marks for guessing!)
Lets now, hit the bulls eye !!
Let's get started.

Topic 4: Introduction to Java

Java is a general-purpose programming language created for developers (to write once and run anywhere) by James Gosling at Sun Microsystems later acquired by Oracle Corporation. The compiled Java code can run on all platforms that support Java. Java applications are compiled to byte code that can successfully run on any Java Virtual Machine (JVM). Syntax of Java is similar to C/C++. Java works on class-based and object-oriented approach of programming.
It is one of the most widely used programming languages for creating desktop and mobile applications as well as big data processing and embedded systems.

Features of Java Programming

1. **An Object-Oriented Language:** Java is an object-oriented language because it is based on the concept of objects and classes (We will explore more about it once we start to code). Here, a complex problem is divided into smaller sets by using concepts of OOP. This makes Java code more understandable (closer to real-life implementation) and reusable. It also improves design and makes the code easier to maintain.
2. **Java is platform-independent:** The Java code written on one platform can run successfully on other platforms without any modifications. It is not compiled into platform-specific machine code but into platform-independent byte code.
3. **Simple and Secure:** Java is designed to be easy to learn. If you understand the basic concepst of OOP, Java would be easy to master. It is one of the simplest languages as it does not have complex concepts like pointers, operator overloading, multiple inheritance, and explicit memory allocation (as we see in C++).
4. **Large Standard Library:** One of the reasons java is so famous is the availability of a huge standard library. To help software program builders like you (in future!), the Java environment carries and supports loads of classes and methods put into numerous packages.

Why is Java better than other programming languages?
Java has added advantages over other languages and environments that make it very appropriate for almost all programming errands. A few of them are :

1. Java syntax is simple to understand and easy to memorize since it is intuitive.
2. Java has been designed to be simpler to type,compile, debug, and learn than other programming languages.
3. Java is secure, convenient, viable and comes with superior high-level concurrency devices. Because of this, java is regularly the default choice for scientific applications, including natural language processing.
4. One of the coolest things about Java is its futuristic approach. When the whole world today is still stuck in Java 8 or Java 11, Oracle has already launched Java 18. The whole development community associated with java is very progressive which has kept it up to date since its inception.

Topic 5: Architecture

Before moving ahead with programming, we need to understand the architecture of Java with its components.

Java Architecture comprises of three major components viz. **JVM, JRE,** and **JDK.**
Let us discuss each of these in brief.

JVM:
Java Virtual Machine or JVM, loads, verifies and executes Java bytecode (object code which is further converted into machine code). It is known as the interpreter or the core of Java programming language because it executes Java programming.The JVM manages system memory and provides a portable execution environment for Java-based applications. JVMs are available for many hardware and software platforms (i.e. JVM is platform dependent).
Examples: Hotspot,Eclipse OpenJ9

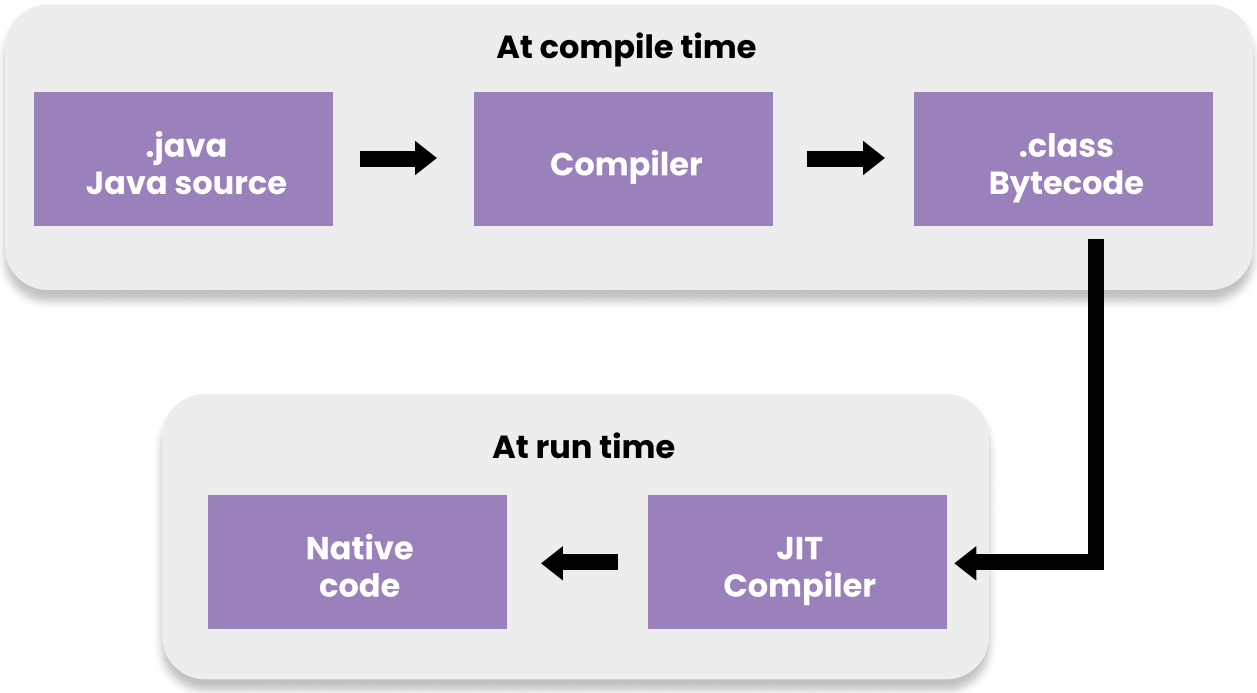
JRE:
The JRE is the underlying technology that communicates between the Java program and the operating system. It acts as a translator and facilitator, providing all the resources so that once you write Java software, it runs on any operating system without further modifications.
A software program needs a runtime environment that provides access to memory and other system resources such as program files and dependencies. In the past, most software used the operating system directly as its runtime environment. However, this meant that developers had to write different code for each operating system that they wanted their applications to run on. The Java Runtime Environment (JRE) technology was created as a solution to this problem.

JDK:
The JDK is a collection of 'software tools' that you can use for developing Java applications. You can set up the JDK in your development environment by downloading and installing it. Select the JDK software version that matches the Java version you want to use.

Still confused ! Let us look at the precise differences between the three. This

JDK	JRE	JVM
JDK is primarily used for code execution and has prime functionality of development.	JRE is majorly responsible for creating an environment for code execution.	Specifies all the implementations and is responsible to provide these implementations to JRE.
JDK = Java Runtime Environment (JRE) + Development tools	JRE = Java Virtual Machine (JVM) + Libraries to run the application	JVM = Only Runtime environment for executing the Java byte code.

Let us now sum it up with a simple block diagram for better and easy understanding.



Conclusively, the overall working of Java can be summed up in the following steps:

- It starts with the process of compilation and interpretation.
- Java compiler converts the Java code into bytecode.
- JVM converts the byte code into machine code.
- The machine code is then executed by the machine/computer and you receive the desired output.

Hope, you now have a basic understanding on how java works.
Absolutely fascinating, isn't it ?!
You wil definitely have more and more clarity once we start with the actual coding/ programing.

See you in the next class !! Keep learning ! Keep exploring !!

Upcoming Class Teaser:

- Java installation guide
- IntelliJ Editor installation