Unit 1

Introduction to JAVA

Java is a general-purpose object-oriented programming language developed by Sun Microsystem of USA in 1991 by the team lead by James Gosling. Initially, it was named as Oak but was renamed Java in 1995. The primary motivation to create java was the need for the platform independent language that could be used to create a software to be embedded in various consumer's electronic devices like microwave, oven, remote control etc. Java is related to C++ i.e., much of the character of Java is inherit from the C++ and C but it is not enhanced version of C++. The change that prompted Java was a need for platform independence program for distribution over internet. The difference in C++ and Java is that java enhanced and refined object-oriented paradigm used by C++, remove complexity and need of separate compiler according to different operating system exist in C++, added integrated support for multithreading and provided library that simplified internet access.

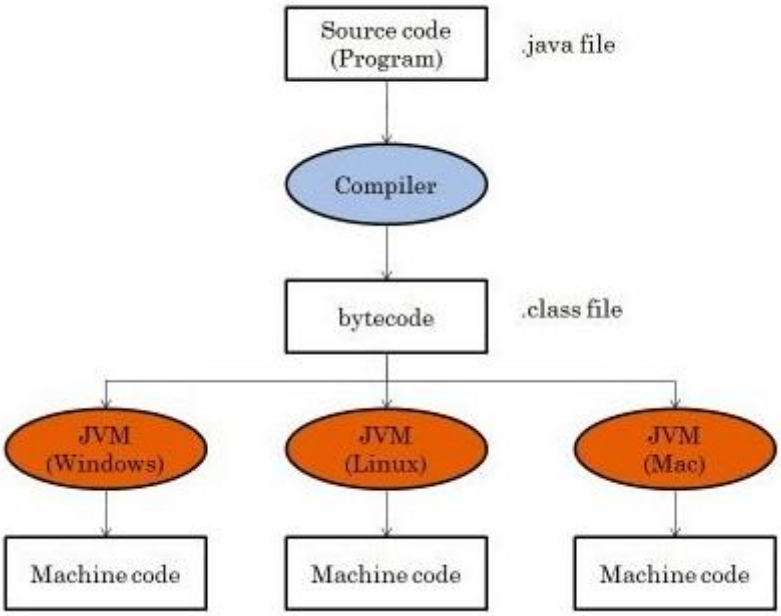
Java is object oriented it means that it everything in java is associated with classes and object along with attributes (data) and method (function). Once a clear well-defined object and reliable data and methods for object is developed then we can simply replace parts of an older system without affecting original features. **Class** is a classification or template of object which encapsulates the data and associated behavior. It gives the blue print or description or structure of the object that can be created from it. **Object** is any entity that has state and behavior. It may have physical existence like a customer, bike, car etc. or conceptual existence like project, process etc. Object are created through class. So, to define the state and what object does or function we need class. For example

Let us consider a motorbike which have attributes like color, brand name, registration number etc. and behavior or functionality like brake, run, signal etc. this provide an overall structure of a specific bike (provide all the mechanism that a bike must have in order to operate) so this is a class. Now, lets consider a bike object B1, which have red color, brand name pulsar, registration number 334, run up to 100km/hr., uses disc brake to stop and use led light to signal. This is the unique structure and behavior of B1 but Bike b1 must have common structure and behavior (brake, run signal) that a particular motorbike has. Therefore, Bike b1 is an object which have structure and behavior inherit from class motorbike i.e., b1 is created from class Motorbike.

Byte Code:

Byte code is a highly optimized set of instructions that represent the task to be executed in execution phase designed to be executed by the java run time system also known as Java Virtual Machine (JVM). It is the intermediate representation of program that contains instructions the java virtual machine will execute. **Bytecode** is produced after java program (file with .java extension) is compiled which makes much easier to run a program in different environment because only JVM is needed to be implemented for each platform. Although the set of instruction for JVM may differ from one platform to another but all JVM can interpret the bytecode. This makes bytecode **platform independent and portable** (without recompiling the source code the same byte code instruction can be execute on any platform that contains JVM). To compile a java program, first the program should be saved on .java format. After this, javac command is used to compile a java file. this will translate a source code into byte code.

Suppose if we have a file name Hello.java then javac Hello.java will compile the file which will produce separate .class called **Hello.class.**



Java Virtual Machine (JVM):

Java Virtual Machine is an abstract machine or an engine that provides run time environment to execute bytecode and converts bytecode into appropriate machine language. JVM are different for different system. So JVM is platform dependent but all JVM can interpret bytecode produce by java compiler.

The JVM is invoked by **java command. For example**

java Hello;

the above command will load .class file into primary memory, verifies the bytecode and executes the bytecode to perform specified action.

Virtual Machine (VM):

Virtual machine is a software application that simulates a computer but hides the underlying operating system and hardware from the programs that interact with it. Because of virtual machine concept we can use two operating system at a time in one system. Virtual machine lures a system that it is using only one OS but in background other OS is operating.

Features of java:

1. **Simple:**
   Java is simple and small language and easier to learn if a person has basic concept on object-oriented programming. Java includes many similar syntaxes of C and C++ and also many similar features of object oriented as that of C++, so people who have knowledge on C and C++ will have little trouble learning java.
2. **Object Oriented:**

Almost everything in java is an object and comes with an extensive set of classes arranged in package. All program code and data reside with object and classes. The object model in java is simple and easy to extend.

3. **Multithreaded:**
   Java supports multithreaded programming which allows to write programs that do many things simultaneously. Due to this feature, we don't need to wait for application to finish one task before beginning of another task. Multithreading breakdowns the whole process into different subprocess and execute at same time. Example of multithread: in MS word we are writing something and in same time it is checking for spelling. While playing Pubg you and your friend run at same time, shoot at same time. This is due to multithreading concept.
4. **Robust:**
   Java has strict compile and run time checking for data type and also incorporates the concepts of exception handling which captures series of errors and eliminates any risk of crashing the system. Java is strictly typed language which checks code at compile time as well as run time which helps to eliminate error prone situation.
5. **Architectural neutral:**
   Java generates architectural neutral object file known as bytecode which is machine independent and recognized by JVM installed in any system. This makes the compiled code executable on many processors hence, can be run on any machine. This makes the java architectural neutral. The main goal of java is "write once run anywhere, anytime, forever".
6. **Interpreted and High Performance:**
   Java performance is impressive due to uses of intermediate bytecode which can translate program into appropriate machine code and also reduces overhead during run time overhead of having separate compiler for separate machine. As byte code is highly optimized the use of bytecode enables the JVM to execute program much faster.
7. **Distributed:**
   Java is designed for the distributed environment of Internet for creating application on networks and handles TCP/IP protocol. Java application can open and access remote object and methods on in internet which enables multiple programmers at multiple remote locations to collaborate and work together on a single project.
8. **Dynamic:**
   Java program carry extensive amount of run time information that is used to verify and resolve accesses to object at run time which makes possible to dynamically link code in safe and expected manner. Java can determine the type of class through a query, making it possible to either dynamically link or abort the program depending on the response.

How java changed the Internet:

Although primary motivation for creation of java was need for platform independent, the emergence and spread of World Wide Web pushed java to design language design for web as web also demanded portable program. To simplify web program, java innovated new type of networked program called applet and also addressed some of the issued associated with internet: portability and security.

Java Applets:

An applet is a special kind of java program that is designed to be transmitted over the internet and automatically executed by java compatible web browser. An applet is downloaded in demand and run-in browser without further interaction with user. Applets are typically a small program used to display data provided by the server, handle user input and also simple function such as calculation that execute locally rather than on the server. The use of applet helps in maintaining security and portability.

Portability:

Portability is a major aspect of the Internet as different types of computer and operating systems are connected to it. As applet is used by java for web, same applet can be downloaded and executed by the different version of operating system, CPU and browser.

Security:

When we download something from browser, we are taking risk as the downloaded program might contains malicious code. But in Java, applet is not allowed to access other parts of the computer i.e., all the request to server, downloaded file will be implemented by applet not on local system. Hence, no any harm will be done on system and security will not be breached.

Java on server side: Servlet

A servlet is a small program that executes on server and extends the functionality of web server which is used to create dynamically generated content that is then served to the client. For example, an online page might use servlet to look up the product and price of product in database. The product and price information are then used to dynamically generate a webpage that is sent to the browser. As servlet are compiled into bytecode and executed by the JVM they are portable, secure and can be used in a different server environment.

Object Oriented Programming:

Object Oriented programming is a programming paradigm which relies on the concept of classes (making the common structure or blueprint) and objects (formed by inheriting common structure) along with attributes (data) and method (function). OOP structure a program into simple, reusable piece of code called class, organize such structured program around its data (object) with a set of well-defined services to that data. Class defines a structure and behavior that will be shared by a set of objects. Each object of a given class shares a structure and behavior defined by a class. Therefore, class is a logical construct and object is a physical reality. Example of class and object is given on page number 1.

Some of the well-known object-oriented languages are C++, Java, Python etc.

Abstraction:

Abstraction refers to the act of representing essential features by hiding the complex details or explanation. This helps to manage complexity. For example: in motor bike we can accelerate to move it, brake to stop it and shifts the gear to control speed. But have we ever thought how brake work, what are its internal components? This is known as abstraction that is we know brake will stop the bike (implement functionality) but we ignore how brake work (hiding internal structure). Hierarchal classification can be used to manage abstraction.

Benefits of OOP:

- One class can be inherited to another class for the common features available without making changes on the previous class. This will help to eliminate redundant code and extend the use of existing classes.
- Emphasis on data rather than procedure
- Data is hidden and cannot be accessed by external function.
- OOP uses abstraction mechanism (hiding the detail and complexity) which helps to filter out limited data to exposure. This helps to provide only necessary data for viewing which cannot be invade by the code in the other part of program.
- Multiple objects can coexist without any interference

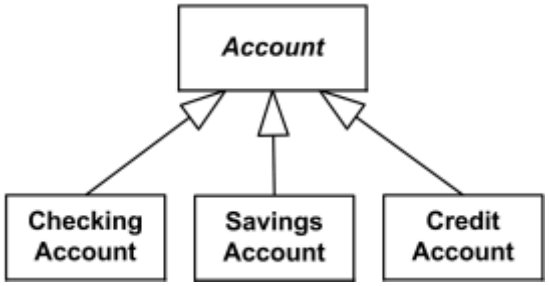
OOP principle:

1. **Encapsulation:**

   Encapsulation is the mechanism that binds together code and the data it manipulates and keeps both safe from outside interference and misuse. Encapsulation can be thought of as protective box that prevents the code and data from being accessed by other code defined outside the box. Access to the data is tightly controlled through well defined interface. The basis of encapsulation is a class. Class contains data and methods that will be shared by a set of objects. Each object of a given class contains structure and behavior define by class. The class can hide the data and method it contains by using the keyword private. This is known as encapsulation. The benefit of encapsulating a data and method is that they are only accessible by a code that is member of class i.e., code that are outside the class cannot access the data. This will help to prevent unauthorized access to class data and methods. Generally, attributes (member variable or data) of class are marked as private. Such private data or member of class can be accessed only through the public methods. Therefore, public interface should be designed carefully such that it does not expose internal working of class.

2. **Inheritance:**

   Inheritance is the process by which object of one class acquires the properties of object of another class or it is the process of acquiring common properties of one class by another class. This supports the concept of hierarchical classification. By use of inheritance an object needs to define only those qualities that make it unique. General (common) attributes are directly inherited from parent class. Let consider following example:



   Here, class account contains the general information like user's account number, demographic information and his/ her balance. The account can be different like saving

account, current account etc. But the general information for all the account is same as that of class account. So, by use of inheritance the different account class like saving, credit and checking does not have to mention general information (account number, balance etc.). the general information is directly inherited or capture from class Account. Therefore, the child class: saving account, credit account and checking account have to mention its unique information such as saving account have own features like bonus 10%, balance retrieval time etc., credit account may have features like badge 2%, credit renewal time etc.

Those class that captures general (common) information are known as **super class**. Those class that inherits common information from superclass and only mention its unique information is known as **sub class**.

3. **Polymorphism:**

Polymorphism (many form) is the quality that allows one interface to access a general class of action. Polymorphism means ability to take more than one form i.e., it is the mechanism from which one operation may exhibit different behavior depending on different instances. The behavior depends on type of data used in the operation. Let consider an example: for addition operation (+): if we provide numeric data, it will perform addition operation and gives sum as output but if we provide string data as input it will perform concatenation operation.

Polymorphism allows an object to have different internal structure but shares the same external structure. Polymorphism is extensively used in implementing inheritance. Polymorphism is often expressed by the phrase "one interface, multiple methods".

Procedure Oriented Programming:

A procedure-oriented programming is a type of programming language in which program is characterized as set of linear step or code i.e., program is written in the form of a sequence of instruction and such code tells the computer to do something i.e., instruct a device on how to finish a task. It focuses on fixed procedure (function) and algorithm is needed to perform the derived computation. This can be thought as code acting on data. Here, procedure consists of some series of computational steps to be carried out which is called while executing the program. Some of the procedure Oriented Program are C, BASIC, Fortan.

Object Oriented Programming vs Procedure Oriented Programming

Object Oriented Programming	Procedure Oriented Programming
OOP structure a program into simple, reusable piece of code called class, organize such structured program around its data (object) with a set of well-defined services to that data.	Procedure oriented characterized program as set of linear step or code and such code tells the computer to do something i.e., instruct a device on how to finish a task.
Programs is divided into small chunks known as object	Main program is divided into small parts based on procedure (function) and treated as separate program for that small parts.

OOP includes access restriction or specifier like public, private, package which prevent from unauthorized access	No any access specifier so prone to data leakage or unauthorize access
Flexible to add new data and methods	Adding new data and function is not easy.
It supports data hiding through interface and access specified hence, more secure	Does not have proper way for data hiding so it is less secure
In OOP data is more important than function	In procedure oriented, function or procedure is more important than data

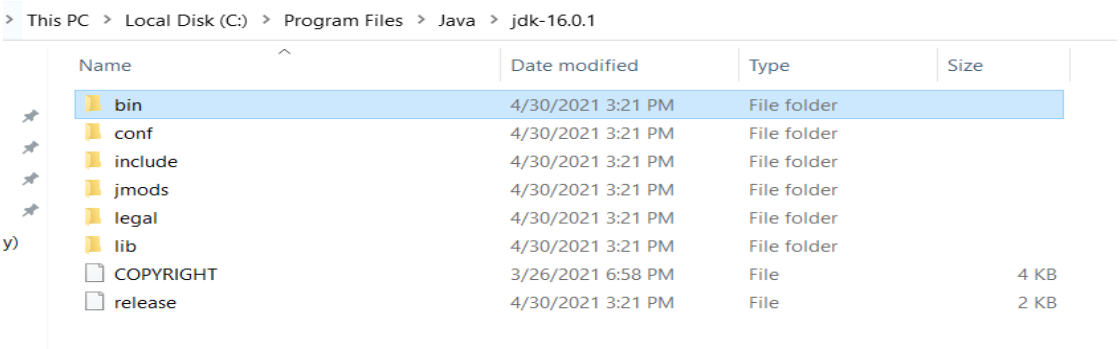
Setting up computer for Java Environment

Step 1: To run a java program first we have to download java development kit (jdk) which consist all the necessary environment like compiler, virtual machine to run java code. After this some text editor like notepad, notepad ++, sublime text is required. Some Integrated Development environment like NetBeans, Eclipse can also be used. For our module we will use NetBeans as IDE.

Step 2: we should create a path environment variable in our system.

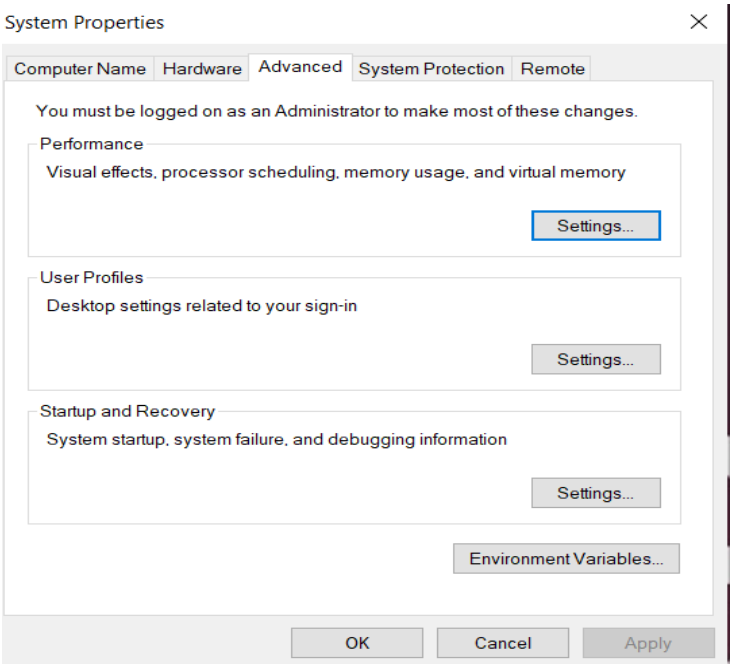
For this go to the location where jdk is installed. In my case jdk is installed in C:\Program Files\Java\jdk-16.0.1.

After this go into bin folder of jdk and copy the path:



In my case path is: C:\Program Files\Java\jdk-16.0.1\bin

After this go to the system properties and select environment variable:



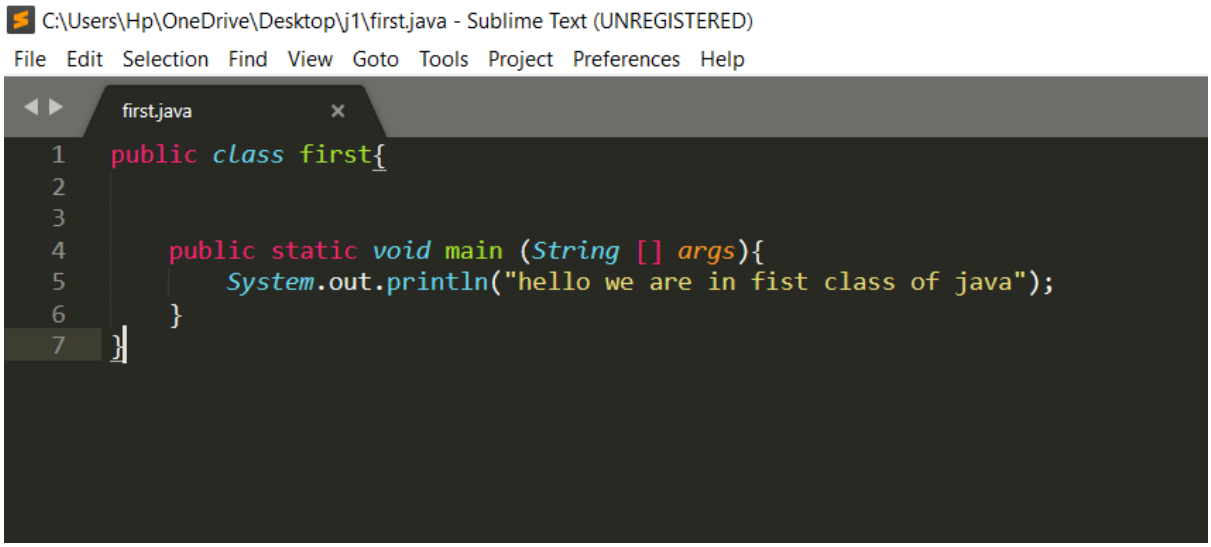
Now add path upto bin folder to the path variable:

After this create new path name JAVA_HOME and insert the path upto jdk folder.

Note: this step is taught in class practically. So, follow the class notes.

Creating a Java Program and Running a Java Program

To run a java program, first we should write a java program in some editor or IDE. In my case I uses sublime text. After writing a java program, save it in appropriate location.

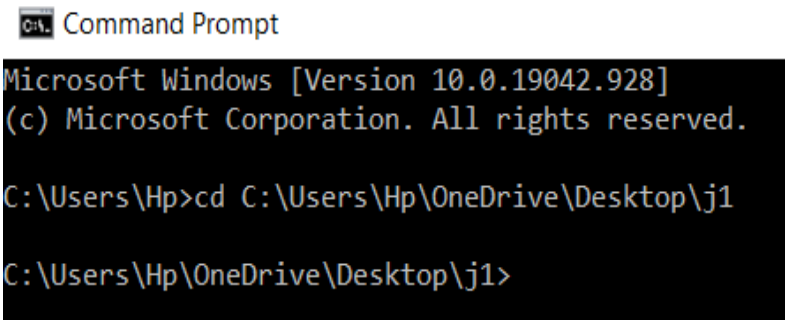


The above program will print “hello we are in first class of java”.

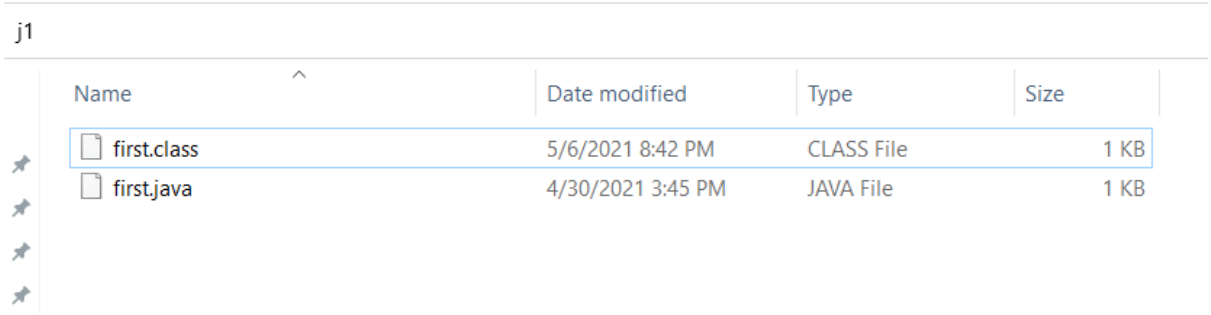
In my case, I save a file in desktop on folder j1. The file name should be same as that of class name and should be saved with .java extension.

After this open command prompt and use following command to compile and run java program:

Step a: go to a location where a java file is saved. In my case path is C:\Users\Hp\OneDrive\Desktop\j1;



Step b: run command **javac name of the file.** in my case javac first.java. this command will create byte code (.class) for the file which will be recognized by our java virtual machine and converts into appropriate machine code. From this command. class file will be created in the location in which java file was resides.



Step c: run command java name of .class file. in my case **java first.** This will invoked JVM, checks the byte code to ensure that they are valid and convert byte code into machine readable code, executes byte code and runs a program