

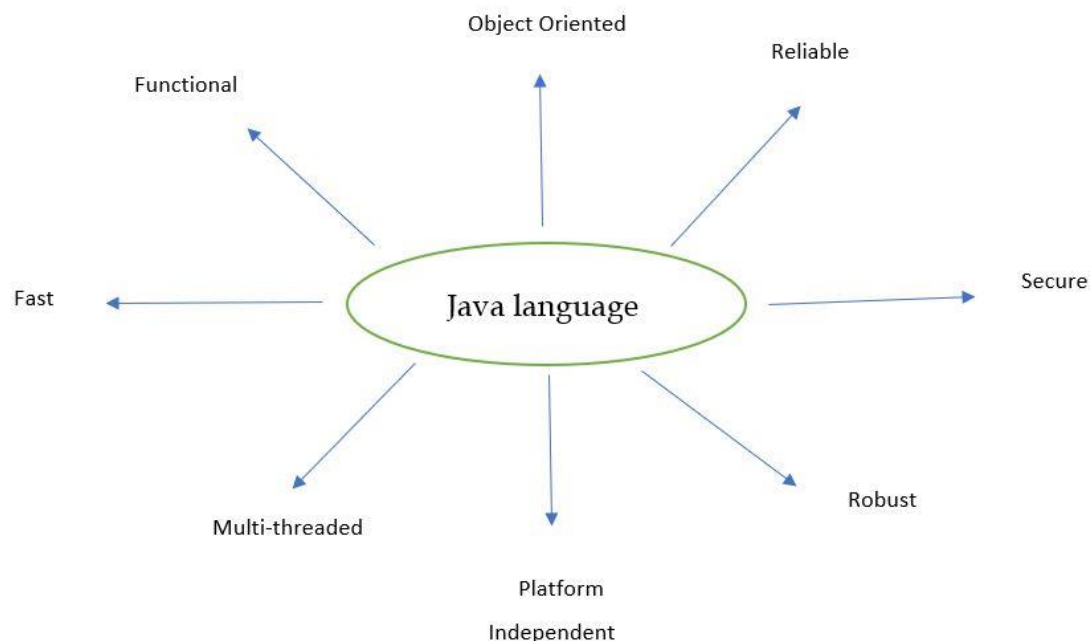
Part 01. Java Fundamental

Lesson 01. What is Java

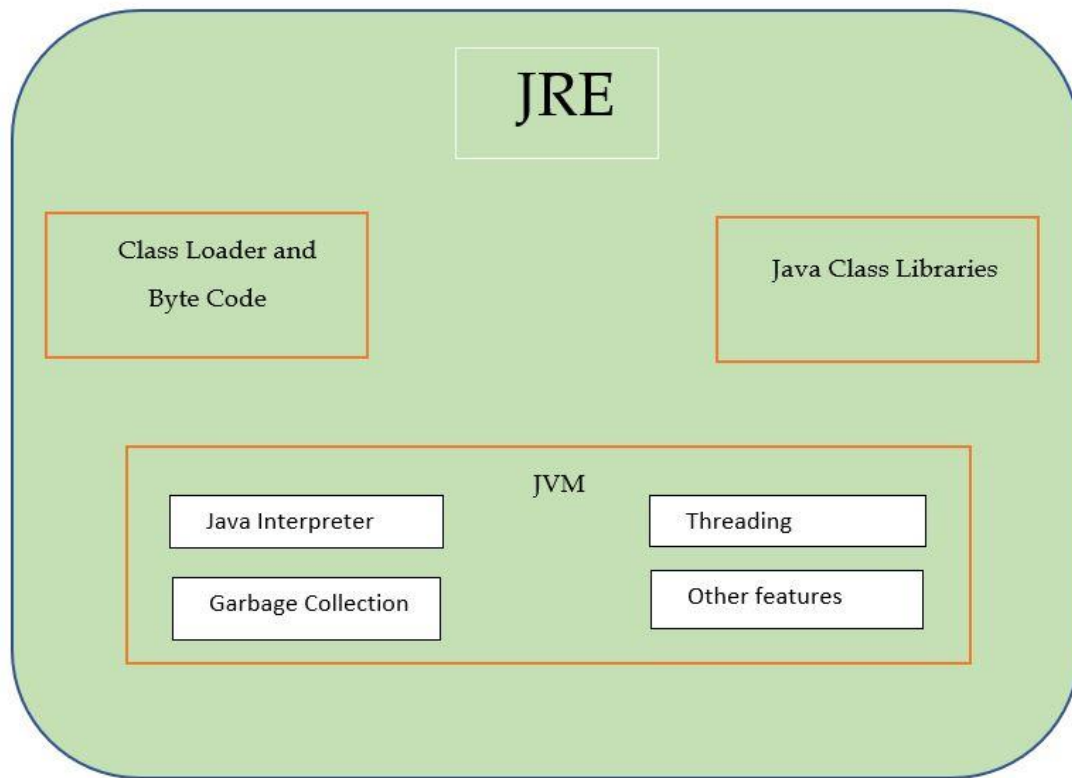
Java can be defined as a fast, secure, and reliable object oriented programming language. Java is also a platform. Now let's understand these two different things in detail.

Java, a programming language – Like many other programming languages like C, C++, python, Visual Basic, JavaScript and many more, Java is also a programming language. Below are the main features of Java

Object Oriented, Fast, Secure, Reliable, Simple, Robust, Platform independent, Multi-thread, concurrent, Functional



Java as a platform – A platform is a piece of software which is designed in such a way that can run different smaller programs. Java comes with its own runtime environment known as JRE on which Java programs can be run. And therefore it is also known as a platform. Java platform consists of runtime components, all required libraries and binaries.



Types of Java Applications: Java applications can be categorized in following

1. **Standalone applications** – These types of applications are created using AWT and Swing. These are mainly desktop applications and need to be installed on every machine which needs to run it. These are also known as Windows applications.
2. **Web applications** – A web application is the one that can be accessed over the web i.e. internet or intranet. Java web technologies like JSP, Servlets, spring, hibernate, jsf, springboot are used to develop web based applications.
3. **Enterprise applications** – These applications are distributed in nature. Now, technologies and services are used across the enterprises in order to enable software applications and hardware systems. Banking applications are examples of enterprise applications.
4. **Mobile applications** – An application that is designed to run on mobile devices such as mobile phone and tablets is known as mobile application. Android and JAVA ME technologies are used for this purpose.

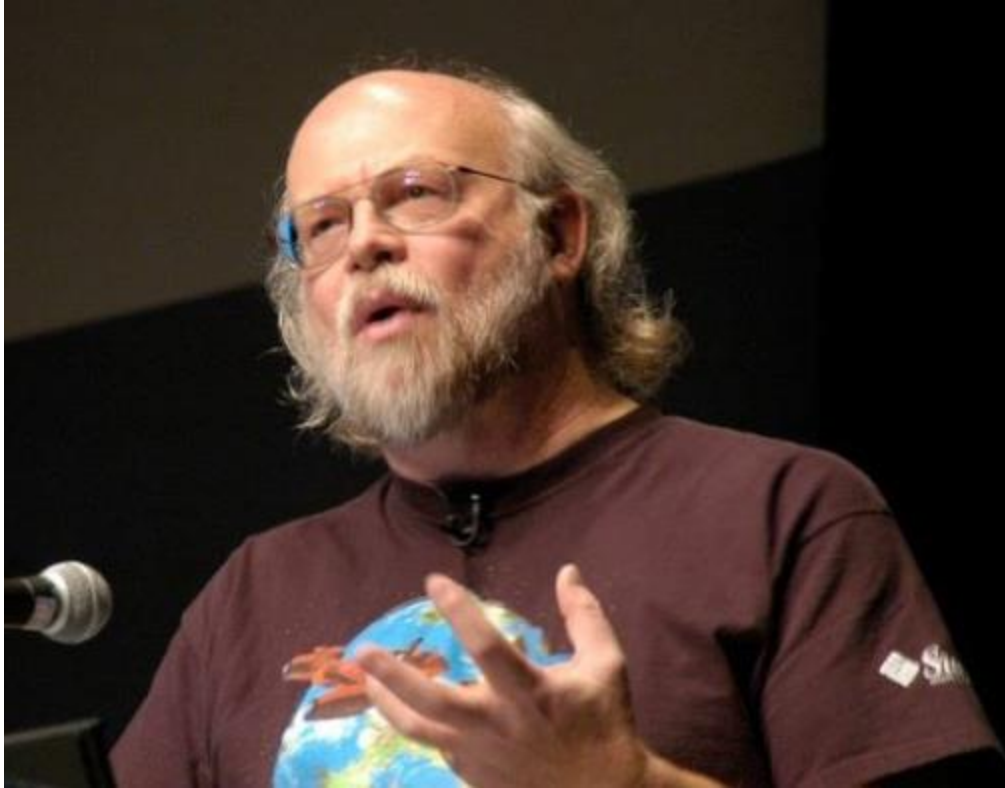
Lesson 02. History of Java

Java has a very interesting story. A team of few engineers was formed in Sun Microsystems to develop a fast, secure and reliable programming language under the leadership of James Gosling. The project was started in June 1991 along with Mike Sheirdan and Patrick Naughton headed by James Gosling. The language was initially designed for interactive television but it proved to be too advanced for cable television industry at that time. However, it was best suited for internet programming. Below are some interesting points about the language as far as its history goes.

- The team of engineers that started working on a new language (today known as Java) was called 'Green Team'.
- Initially, the language was given name 'OAK' after an Oak tree that stood outside Games Gosling's office.
- But this didn't work as there was a name conflict with Oak Technologies. So, the name OAK was dropped.
- Later the project ran by the name 'Greentalk' and the file extension was .gt.
- In 1995, the language was given a name which is continued till date. That name is 'Java'. This is how Java got its name.
- There were multiple choices for the name along with Java like Ruby, DNA, SILK etc. But 'Java' got maximum votes within the team and it got its name.
- SILK was the second most popular choice.
- The name 'Java' came to James Gosling's mind when one day he was having coffee near his office. Java Coffee is a type of coffee from Indonesia.



Java Logo



James Gosling

- In 1995, Java was awarded one of the ten best products in the field of technology by Time magazine.
- First version of java 1.0 was released in 1996, January 23.
- Java is the number 1 developer platform in the world.
- More than ten million developers and 13 billion devices use Java till the point this line is written.

Now, since we know little bit of history of Java, it's time to enjoy your cup of coffee or tea. So, enjoy your drink and happy learning.



Lesson 03. Java Version History

The current stable major version till this article is written is 16. Since the release of Java8, Oracle which now own Java releases every even version in the month of March and odd version in the month of September. Following is the list of all the released versions:

Version	Release Year
JDK Alpha and Beta	1995
JDK 1.0	23rd Jan 1996
JDK 1.1	19th Feb 1997
J2SE 1.2	8th Dec 1998
J2SE 1.3	8th May 2000
J2SE 1.4	6th Feb 2002

J2SE 5.0	30th Sep 2004
Java SE 6	11th Dec 2006
Java SE 7	28th July 2011
Java SE 8	18th Mar 2014
Java SE 9	21st Sept 2017
Java SE 10	20th Mar 2018
Java SE 11	25th Sept 2018
Java SE 12	19th Mar 2019
Java SE 13	17th Sept 2019
Java SE 14	17th Mar 2020
Java SE 15	15th Sept 2020
Java SE 16	16th Mar 2021
Java SE 17	14th Sept 2021

Lesson 04. First Java Program

It's time for us to create our first java program and run it. But before that there are certain requirements that we need to meet in order to run our first java program.

Requirements:

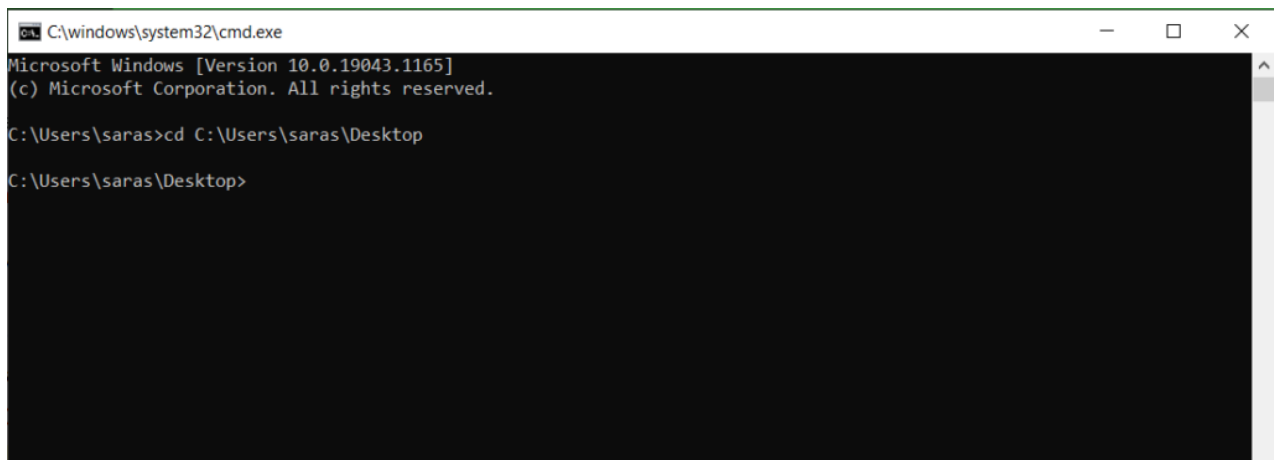
1. You need to have JDK installed on our machine. If you don't have, then download from below link. <https://www.oracle.com/java/technologies/javase-downloads.html>
2. Set environment variable named 'path' and mention the path of jdk bin directory. [Click here to set the 'path' environment variable.](#) If path is already set for java, skip this step.

First Java Program:

1. Open a text editor of your choice and write the following program.

```
public class MyFirstJavaProgram {  
  
    public static void main (String [] args){  
        System.out.println("Welcome to Java programming language");  
    }  
}
```

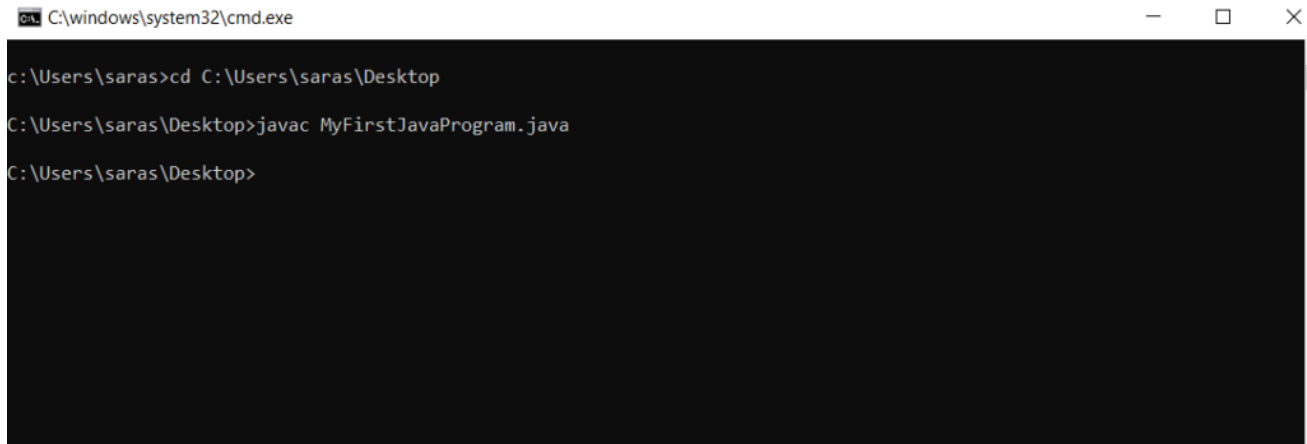
2. Save the file with the same name as class name i.e. MyFirstJavaProgram.java
3. Open command prompt and go to the directory (using cd command) where you have saved your java file. In the below example, the file is saved on desktop therefore we have changed the directory to Desktop .



The screenshot shows a Windows Command Prompt window titled "C:\windows\system32\cmd.exe". The text inside the window reads: "Microsoft Windows [Version 10.0.19043.1165] (c) Microsoft Corporation. All rights reserved. C:\Users\saras>cd C:\Users\saras\Desktop C:\Users\saras\Desktop>".

4. Run the below command

`javac MyFirstJavaProgram.java`



```
C:\windows\system32\cmd.exe

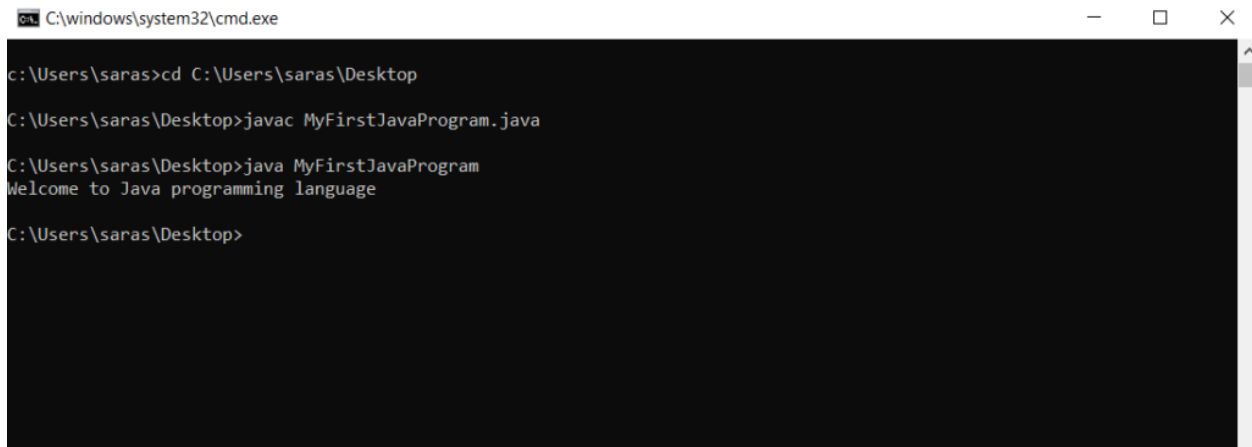
c:\Users\saras>cd C:\Users\saras\Desktop

C:\Users\saras\Desktop>javac MyFirstJavaProgram.java

C:\Users\saras\Desktop>
```

5. Run the below command

```
java MyFirstJavaProgram
```



```
C:\windows\system32\cmd.exe

c:\Users\saras>cd C:\Users\saras\Desktop

C:\Users\saras\Desktop>javac MyFirstJavaProgram.java

C:\Users\saras\Desktop>java MyFirstJavaProgram
Welcome to Java programming language

C:\Users\saras\Desktop>
```

As we can see in the above diagram, the message that we had written in our java program “Welcome to Java programming language” has been printed on the command prompt.

Congratulations! You have successfully written and run your first java program. It’s coffee time now. You deserve your cup of coffee. Enjoy!



Lesson 05. Local Environment Set-up

Now, since you have decided to be a java developer and write some programs, it is time to set up your local environment. Follow the below steps to do so:

1. Install JDK from <https://www.oracle.com/java/technologies/javase-downloads.html>
2. Install any IDE of your choice : Eclipse, NetBeans, STS, IntelliJ Idea, MyEclipse etc. Any of the IDEs will work.
3. Download Eclipse from [here](#)
4. Download NetBeans from [here](#)
5. Download STS from [here](#)
6. Set environment variable 'Path' to include java bin directory. [Click here](#) to know how to set up environment variable.

If you want to write your java programs using a plain text editor, notepad++ can be a good choice for you.

Download notepad++ from [here](#)

Now, you have necessary arrows in your quiver to start. So, be ready to be a strong java developer. It's time for a cup of coffee. Enjoy!

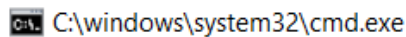


Lesson 06. How to set 'Path' env variable

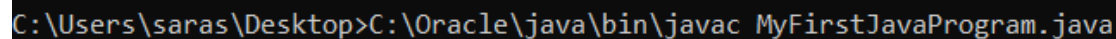
Before knowing how to set 'Path' environment variable for Java executables, we need to know why is it mandatory? The answer is – it is not mandatory but it makes it easier for us to run java programs and some of the other utilities which use JAVA_HOME environment variables to run.

Path variable is one of the environment variables in windows that has comma separated values. In order to execute any utility or executable program such as java, mvn from command prompt, we need to tell windows where that executable file exist. The location of that executable is required to be mentioned in 'Path' variable.

If we do not mention the path doesn't mean we cannot run a java program. We still can, but then we have to give the complete path of java.exe or javac.exe file like below



C:\windows\system32\cmd.exe

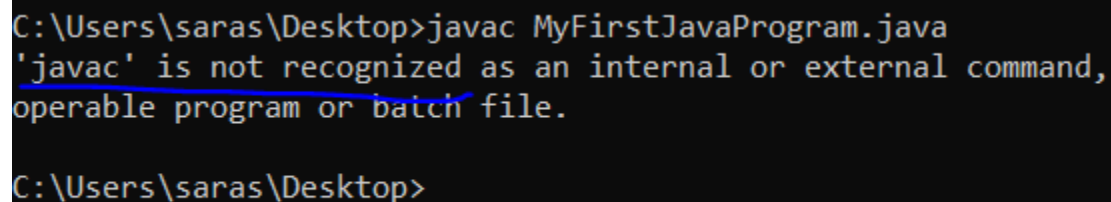


```
C:\Users\saras\Desktop>C:\Oracle\java\bin\javac MyFirstJavaProgram.java
```

Here we have given complete path of javac.exe

But this is not a handy approach. We have to give complete path everytime we want to run a command. That's why set environment variable once and we are good to run all java commands until the machine is formatted or the environment variable 'Path' is mistakenly modified.

Quiz: what happens when neither path variable is set nor complete path is given? Find answer in the below screenshot:

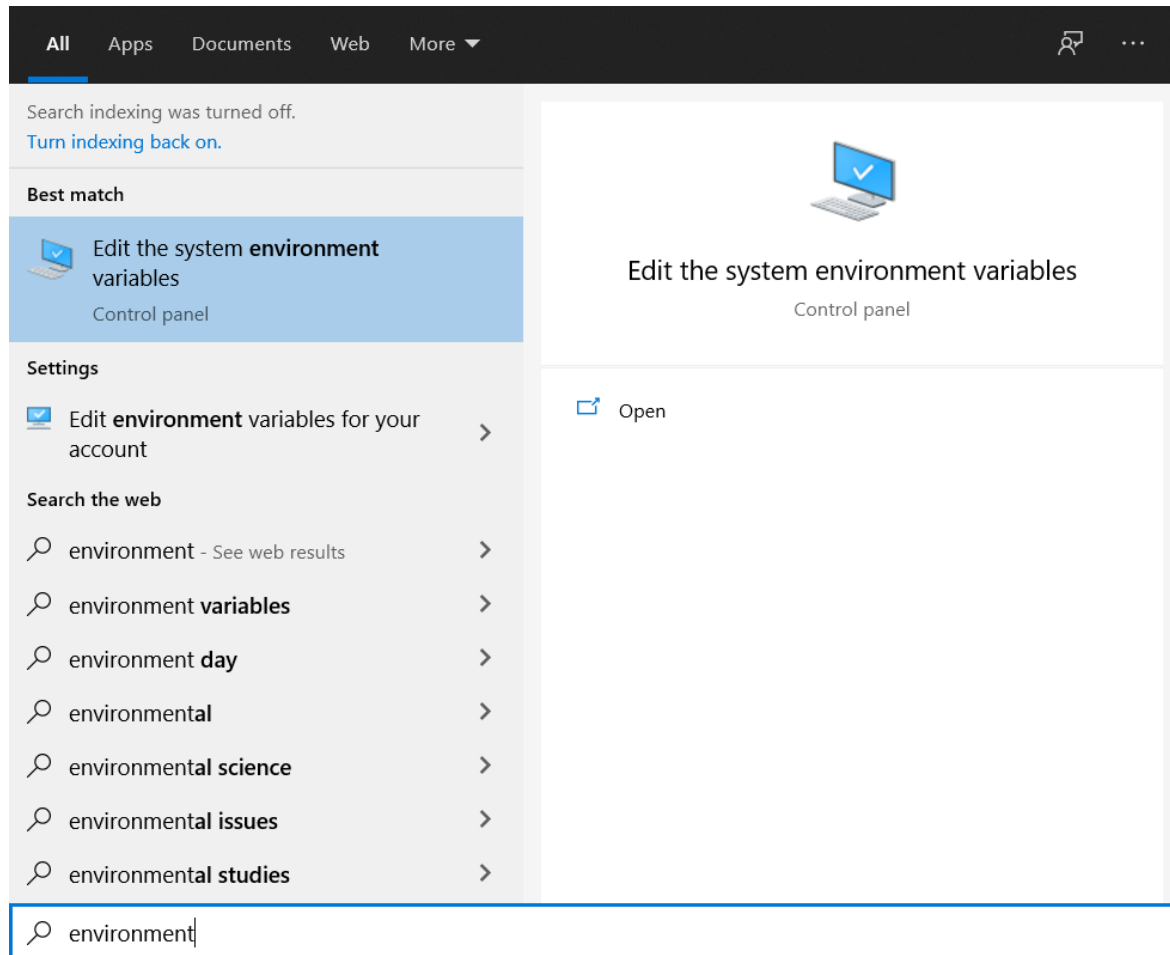


```
C:\Users\saras\Desktop>javac MyFirstJavaProgram.java
'javac' is not recognized as an internal or external command,
operable program or batch file.

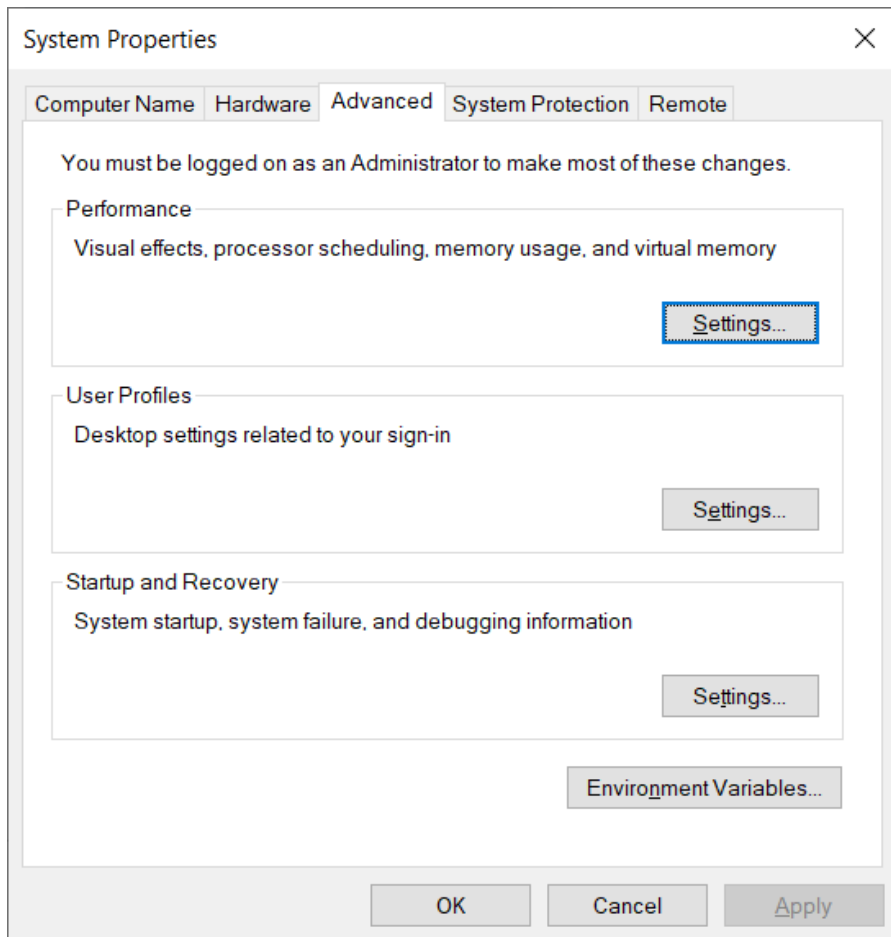
C:\Users\saras\Desktop>
```

Steps to set 'Path' environment variable

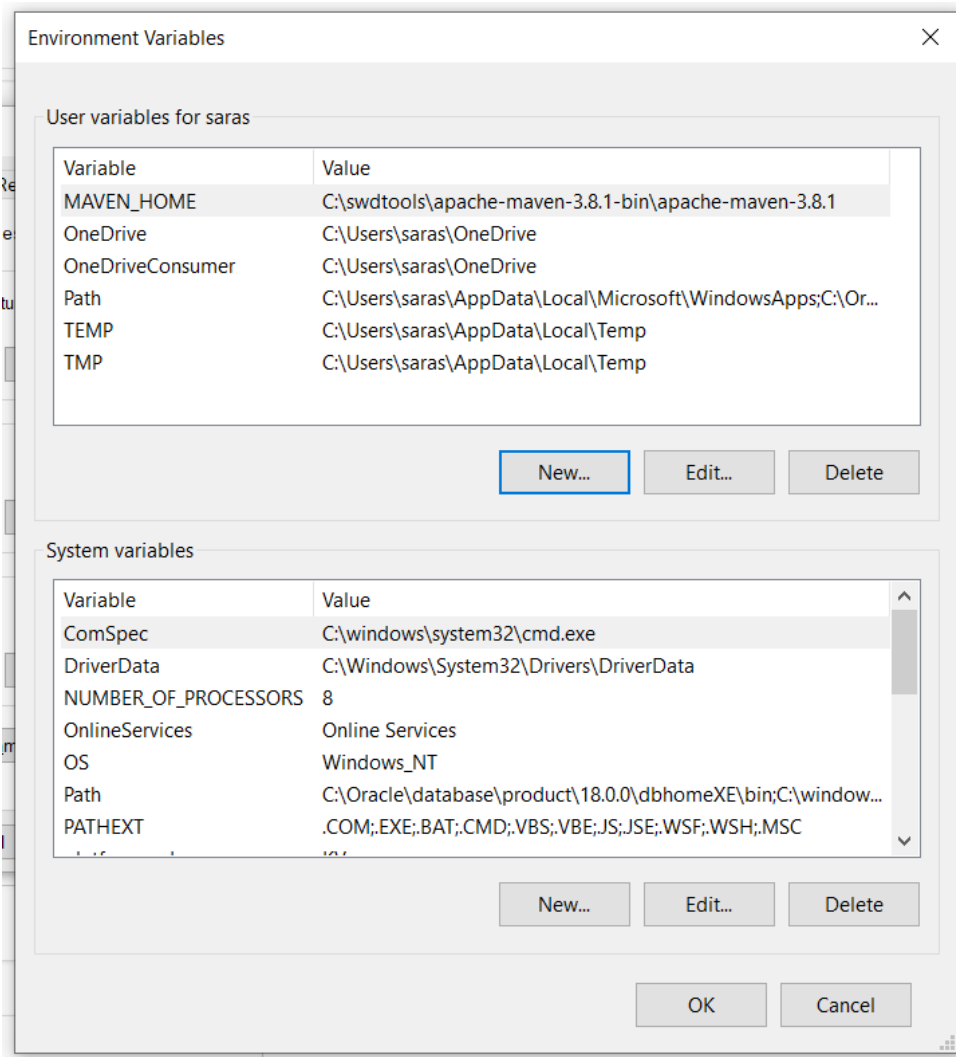
1. Click on the windows button and start typing 'environment..' Below window will appear. Click on '**Edit the system environment variables**'



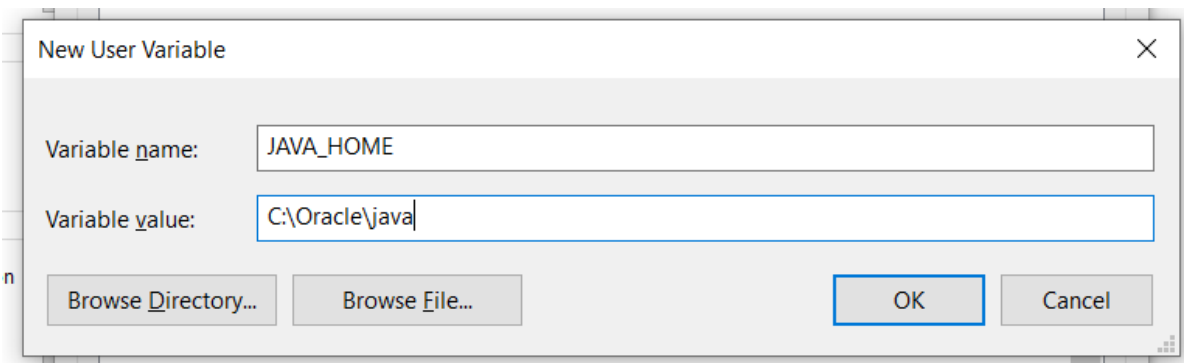
2. In the below window, click on '**Environment Variables...**' button.



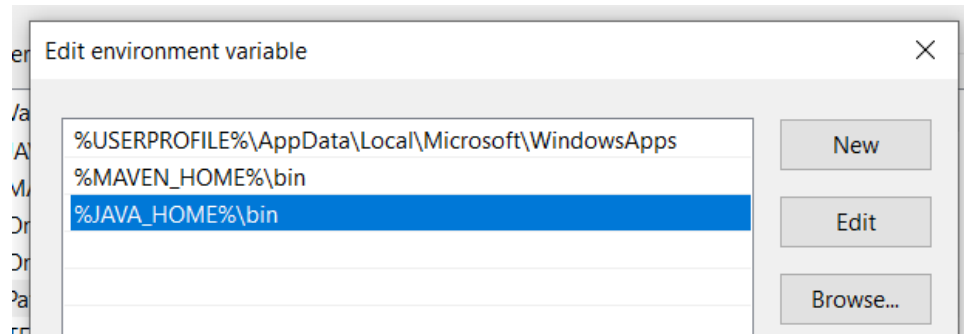
3. Following window will open, click on 'New' button



4. First, we will set a new variable named as 'JAVA_HOME'. Its value would be the location where jdk is installed. In this case, jdk is installed in a folder c:\Oracle\java



5. This would be added to the list as shown below



8. Save these settings by clicking 'OK' button wherever applicable. This is how we have set 'Path' variable for Java. Same steps can be followed to include maven, oracle etc in the 'Path'.

Note -> Using the above steps we have set the 'Path' variable permanently. We can also set it temporarily. The steps are given below.

How to set 'Path' environment variable temporarily

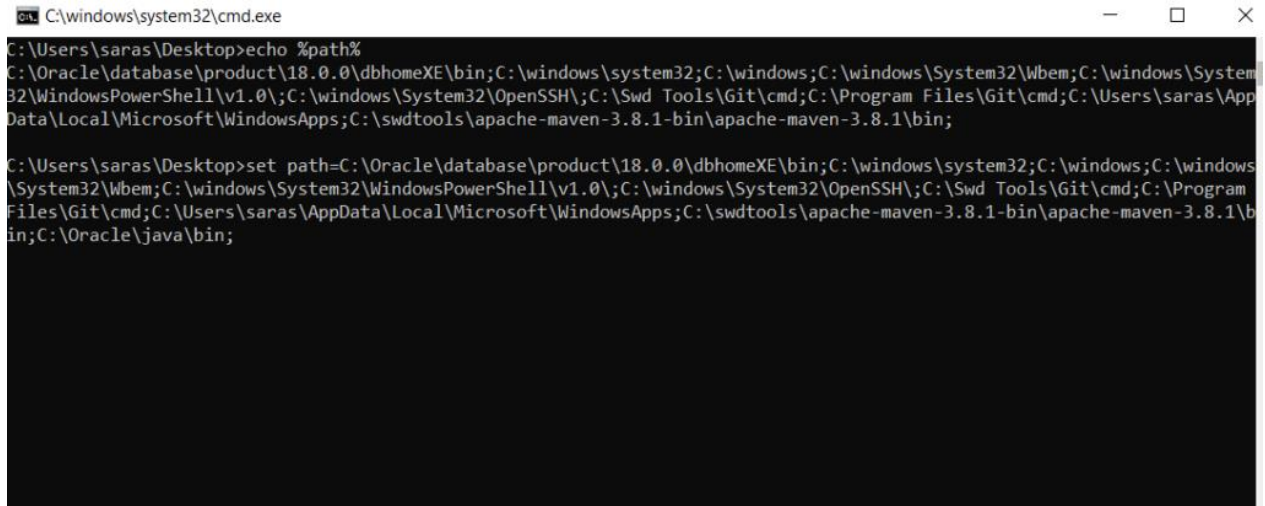
Temporary 'Path' can be set on one command prompt session. Once the command prompt is closed, the original setting of 'Path' will be in effect.

1. Open command prompt and run below command: `echo %path%`

```
C:\windows\system32\cmd.exe
C:\Users\saras\Desktop>echo %path%
C:\Oracle\database\product\18.0.0\dbhomeXE\bin;C:\windows\system32;C:\windows;C:\windows\System32\Wbem;C:\wind
32\WindowsPowerShell\v1.0\;C:\windows\System32\OpenSSH\;C:\Swd Tools\Git\cmd;C:\Program Files\Git\cmd;C:\Users
Data\Local\Microsoft\WindowsApps;C:\swdtools\apache-maven-3.8.1-bin\apache-maven-3.8.1\bin;
C:\Users\saras\Desktop>
```


2. Copy the echoed path to a notepad or any other text editor and add path to your jdk\bin directory. And run the below command (as mentioned in the below screenshot)

set path = <existing-path>;C:\Oracle\java\bin;



```
C:\windows\system32\cmd.exe
C:\Users\saras\Desktop>echo %path%
C:\Oracle\database\product\18.0.0\dbhomeXE\bin;C:\windows\system32;C:\windows;C:\windows\System32\Wbem;C:\windows\System32\WindowsPowerShell\v1.0\;C:\windows\System32\OpenSSH\;C:\Swd Tools\Git\cmd;C:\Program Files\Git\cmd;C:\Users\saras\AppData\Local\Microsoft\WindowsApps;C:\swdtools\apache-maven-3.8.1-bin\apache-maven-3.8.1\bin;

C:\Users\saras\Desktop>set path=C:\Oracle\database\product\18.0.0\dbhomeXE\bin;C:\windows\system32;C:\windows;C:\windows\System32\Wbem;C:\windows\System32\WindowsPowerShell\v1.0\;C:\windows\System32\OpenSSH\;C:\Swd Tools\Git\cmd;C:\Program Files\Git\cmd;C:\Users\saras\AppData\Local\Microsoft\WindowsApps;C:\swdtools\apache-maven-3.8.1-bin\apache-maven-3.8.1\bin;C:\Oracle\java\bin;
```

In this way, 'Path' variable is set, but it is applicable only for this command prompt window.

Object Oriented Programming

Java is an Object Oriented Programming language. OOP (an abbreviation for Object Oriented Programming) is a concept which is built around objects. All the programming is done around objects and the actual logic processing is done by the objects.

Java follows this concept of Object Oriented programming. Sometimes, it is asked whether Java is a fully Object Oriented Programming language or not. To answer this question, first, we need to understand what is a fully Object Oriented Programming language. A language which represents all kinds of data only using objects is a fully Object Oriented language. The answer to the question – whether Java is fully object oriented or not – is therefore 'NO', because some of the data is also represented by primitive variables. Primitive data types like int, boolean are not objects, and that's why Java is not fully OOP language.

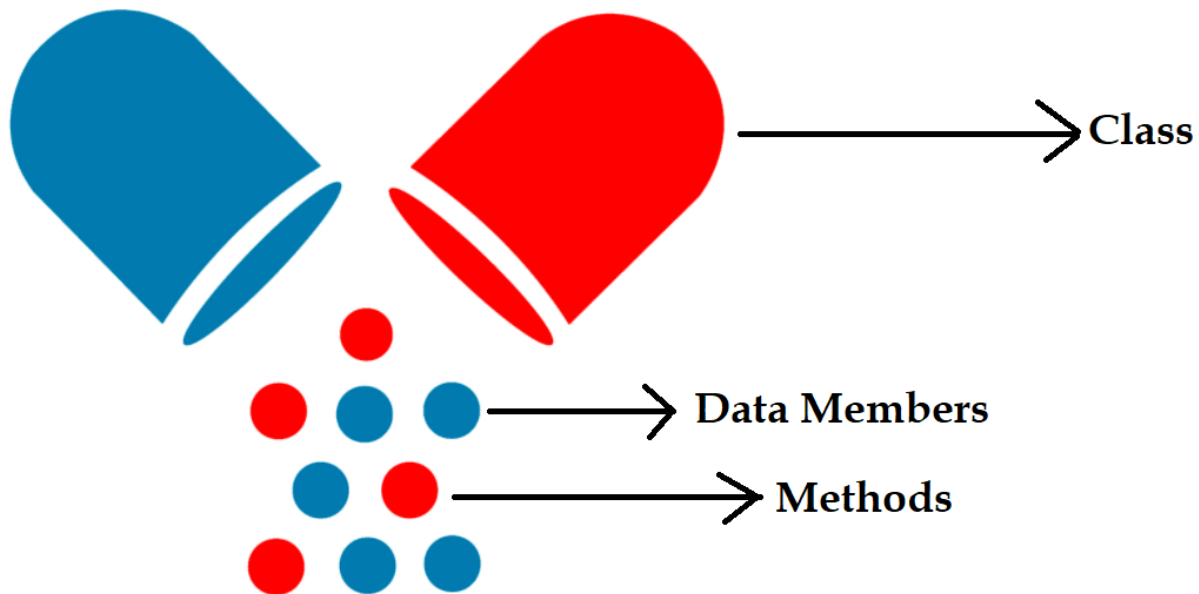
OOP concepts

There are four concepts of Object Oriented Programming:

1. Encapsulation
2. Abstraction
3. Inheritance
4. Polymorphism

Encapsulation

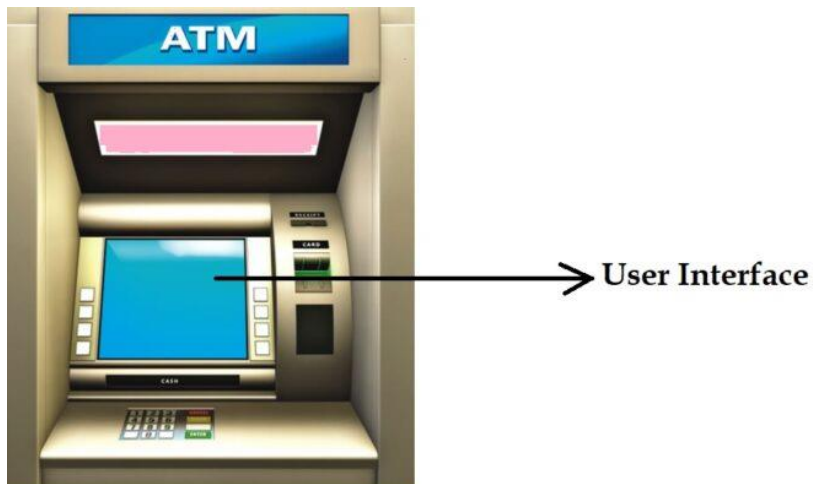
The literal meaning of encapsulation is encasing something in (as if in a capsule). Like medicines are encased in a capsule so that it remain intact and protected. Similarly, in Object oriented programming, data needs to be protected and encased within an entity. That entity, in Java, is known as class. Class helps us implement encapsulation in Java.



Abstraction

Abstraction means hiding the complexity and giving a rather easy interface to the end user of a product. Let's take an example of an ATM machine. It's a huge machine with a lot of processing that goes inside it. But for the end user, there is only a small interface and that is the screen of the ATM. User just need to insert his/her card, enter the correct pin and then choose from the available services. What algorithm is applied inside the machine and what all processing happens inside it, the user is not really bothered. He is only concerned with the output of his transaction. If he has chosen withdrawl of some money, he should get it.

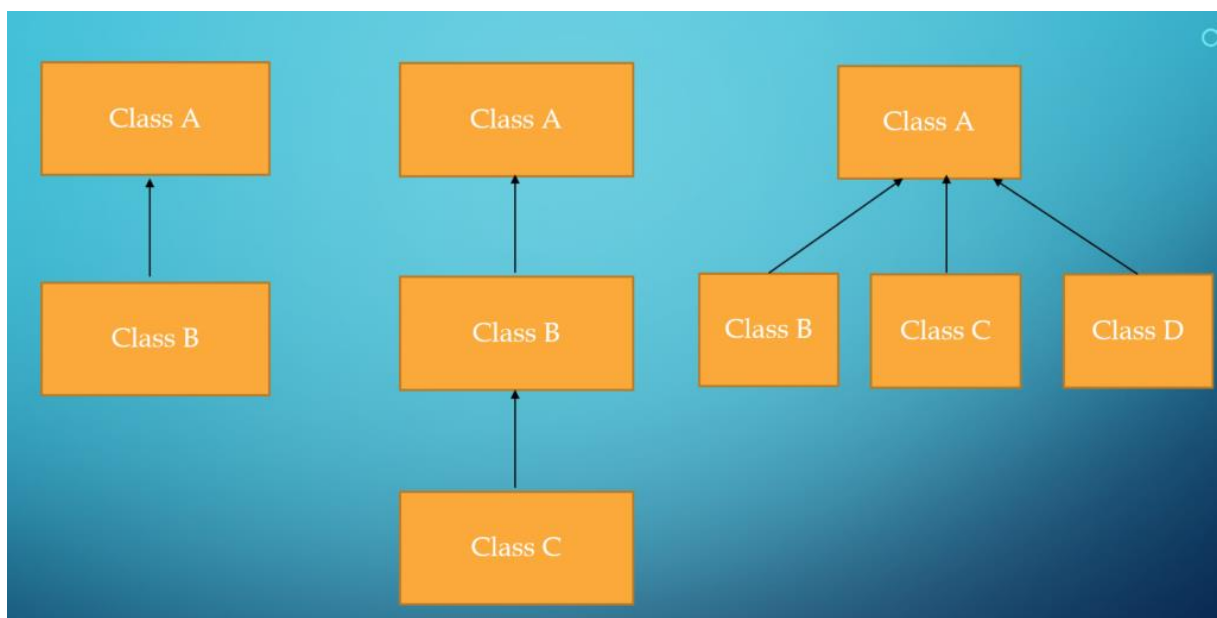
This is an example of abstraction. The entire working of the machine has been abstracted from the user and an easy interface (touch screen) has been provided to him to do his transactions. In java, abstraction is achieved by two ways: **Interfaces** and **abstract classes**.



Inheritance

Inheritance is the action of inheriting some or all the properties from one's ancestors. Human beings inherit properties from their forefathers, like one may inherit basic nature of his/her father. Another person may inherit features of his mother. Some people might inherit some other properties from their father's father and so on. This is inheritance.

In java, inheritance is achieved using parent-child class relationship and interface-(implementing class) relationship. Below it is explained using diagrams:



Inheritance examples

In the first example, Class A is the parent class of class B. In the second example, Class A is the parent class of class B, and class B is parent class of class C. Class A is also parent class of class C, though indirectly.

In the third example, class B, C, and D are the child classes of class A. Using the inheritance principle, all the child classes will inherit all public properties and behaviour of the parent class. Thus, in the second example, class C will inherit public properties of its parent class B as well as indirect parent class A.

We will learn about inheritance in detail in separate chapter.

Polymorphism

Polymorphism means occurrence of many forms in a compound or in a family of living beings. For example, diamond and graphite are two forms of carbon. Therefore, diamond and graphite are called polymorphs of carbon. And this phenomenon is known as polymorphism.

Another real world example is, an individual can have multiple forms in her life. Such as a woman is a mother at home, she takes form of an employee in an organization, she takes form of a responsible citizen of the country at public places and so on. So, the individual is same but in different conditions she takes different forms.

In java, polymorphism is implemented using two ways: Method Overloading and Method Overriding.

To learn more about Method Overloading and Method Overriding, [click here](#).



Polymorphism in Java