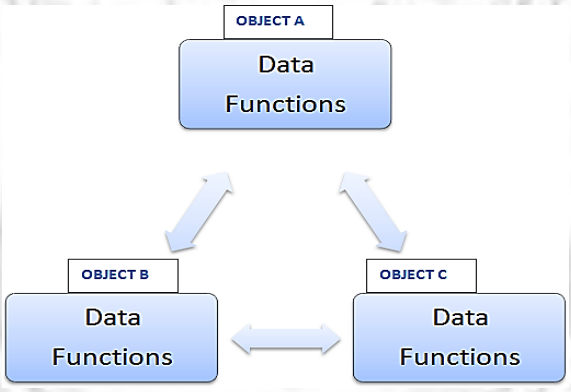
OVERVIEW OF OOP

OBJECT ORIENTED PROGRAMMING (OOP):

Object-Oriented Programming System (OOPs) is a programming paradigm based onthe concept of —objects that contain data and methods, instead of just functions and procedures.

- ✓ The primary **purpose** of object-oriented programming is to increase the flexibility and maintainability of programs.
- ✓ Object oriented programming brings together data and its behavior (methods) into a single entity (object) which makes it easier to understand how a program works.



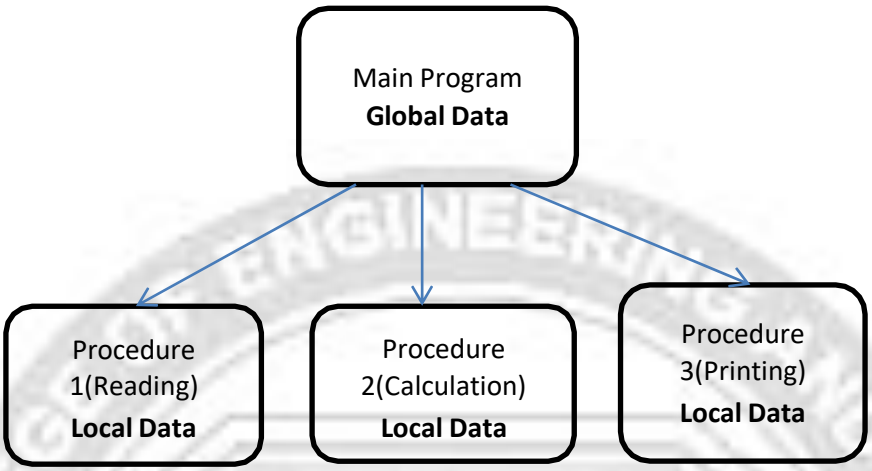
- **Features / advantages of Object Oriented Programming :-**
  1. It emphasis in own data rather than procedure.
  2. It is based on the principles of inheritance, polymorphism, encapsulation and dataabstraction.
  3. Programs are divided into objects.
  4. Data and the functions are wrapped into a single unit called class so that data ishidden and is safe from accidental alternation.
  5. Objects communicate with each other through functions.
  6. New data and functions can be easily added whenever necessary.
  7. Employs bottom-up approach in program design.

PROCEDURE-ORIENTED PROGRAMMING [POP]:

Procedure-Oriented Programming is a conventional programming which consists of writing a list of instructions for the computer to follow and organizing these instructions into groups known as Functions (or) Procedures (or) subroutines (or)Modules.

Example: A program may involve the following operations:

- ✓ Collecting data from user (Reading)
- ✓ Calculations on collected data (Calculation)
- ✓ Displaying the result to the user (Printing)



Characteristics of Procedural oriented programming:-

1. It focuses on process rather than data.
2. It takes a problem as a sequence of things to be done such as reading, calculating and printing. Hence, a number of functions are written to solve a problem.
3. A program is divided into a number of functions and each function has clearly defined purpose.
4. Most of the functions share global data.
5. Data moves openly around the system from function to function.
6. Employs top-down approach in program design.

Drawback of POP

- Procedural languages are difficult to relate with the real world objects.
- Procedural codes are very difficult to maintain, if the code grows larger.
- Procedural languages do not have automatic memory management as like in Java. Hence, it makes the programmer to concern more about the memory management of the program.
- The data, which is used in procedural languages, are exposed to the whole program. So, there is no security for the data.
- Examples of Procedural languages :
  - BASIC
  - C
  - Pascal
  - FORTRAN

- **Difference between POP and OOP:**

	Procedure Oriented Programming	Object Oriented Programming
Divided Into	In POP, program is divided into smallparts called **functions**.	In OOP, program is divided into partscalled **objects**.
Importance	In POP, Importance is not given to **data** but to functions as well as **sequence** of actions to be done.	In OOP, Importance is given to the data rather than procedures or functions because it works as a **real world**.
Approach	POP follows **Top Down approach**.	OOP follows **Bottom Up approach**.
Access Specifiers	POP does not have any accessspecifier.	OOP has access specifiers named Public, Private, Protected, etc.
Data Moving	In POP, Data can move freely fromfunction to function in the system.	In OOP, objects can move and communicate with each other throughmember functions.
Expansion	To add new data and function in POPis not so easy.	OOP provides an easy way toadd newdata and function.
Data Access	In POP, Most function uses Global data for sharing that can be accessed freely from function to function in the system.	In OOP, data cannot move easily from function to function, it can be kept public or private so we can control the access of data.
Data Hiding	POP does not have any proper wayfor hiding data so it is **less secure**.	OOP provides Data Hiding so provides **more security**.
Overloading	In POP, Overloading is not possible.	In OOP, overloading is possible in the form of Function Overloading and Operator Overloading.
Examples	Examples of POP are: C,VB, FORTRAN, and Pascal.	Examples of OOP are: C++, JAVA, VB.NET, C#.NET.

FEATURES / CHARACTERISTICS OF OBJECT ORIENTED PROGRAMMING CONCEPTS

OOPs simplify the software development and maintenance by providing some concepts:

1. Class - Blue print of Object
2. Object - Instance of class
3. Encapsulation - Protecting our data
4. Polymorphism - Different behaviors at different instances
5. Abstraction - Hiding irrelevant data
6. Inheritance - An object acquiring the property of another object

1. Class:

A class is a collection of similar objects and it contains data and methods that operate on that data. In other words – **Class is a blueprint or template for a set of objects that share a common structure and a common behavior**. It is a logical entity.

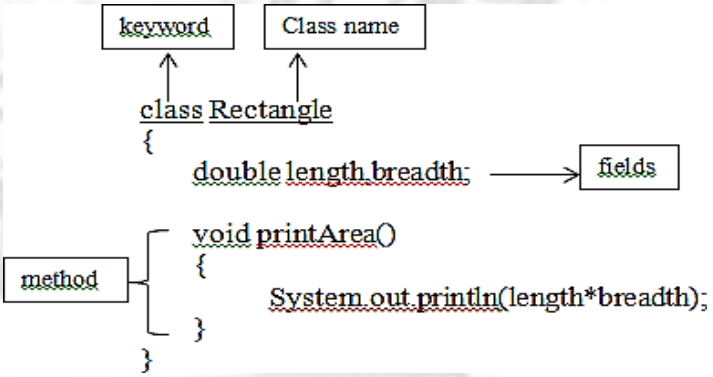
A class in Java can contain:

- **fields**
- **methods**
- **constructors**
- **blocks**
- **nested class and interface**

Syntax to declare a class:

Example:

```
class <class_name>
{
   field;
  method;
}
```



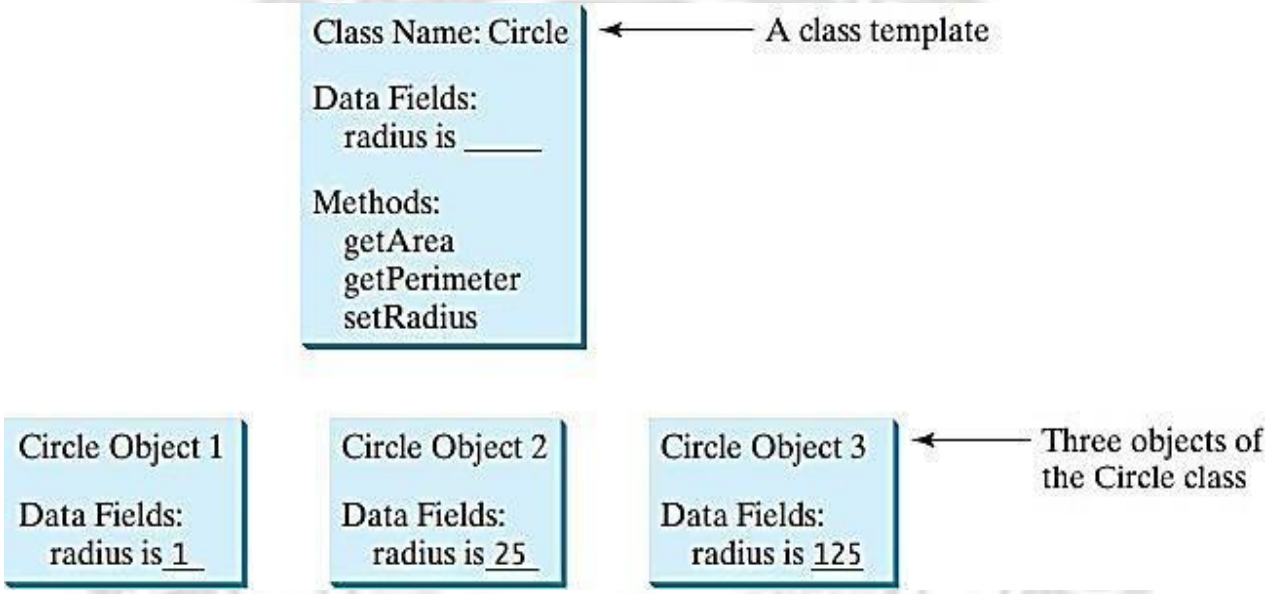
2. Object:

Any entity that has state and behavior is known as an object. **Object is an instance of a class.**

- ✓ For example: chair, pen, table, keyboard, bike etc. It can be physical and logical.
- ✓ The object of a class can be created by using the **new** keyword in Java Programming language.

```
class_name object_name = new class_name;
                (or)
class_name object_name;
object_name = new class_name();
```

Syntax to create Object in Java:



An object has three characteristics:

- **State:** represents data (value) of an object.
- **Behavior:** represents the behavior (functionality) of an object such as deposit, withdraw etc.
- **Identity:** Object identity is an unique ID used internally by the JVM to identify each object uniquely.
- For Example: Pen is an object. Its name is Reynolds, color is white etc. known as its state. It is used to write, so writing is its behavior.

Difference between Object and Class

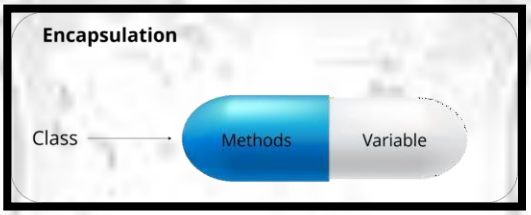
S.No.	Object	Class
1)	Object is an **instance** of a class.	Class is a **blueprint or template** from whichobjects are created.
2)	Object is a **real world entity** such aspen, laptop, mobile, bed, keyboard, mouse, chair etc.	Class is a **group of similar objects**.
3)	Object is a **physical** entity.	Class is a **logical** entity.
4)	Object is created through **new keyword** mainly e.g. Student s1=new Student();	Class is declared using **class keyword** e.g. class Student{}
5)	Object is created **many times** as per requirement.	Class is declared **once**.

6)	Object **allocates memory when it iscreated**.	Class **doesn't allocated memory when it is created**.
7)	There are **many ways to create object** in java such as new keyword, newInstance() method, clone() method, factory method and deserialization.	There is only **one way to define class** in java using class keyword.

3. Encapsulation:

Wrapping of data and method together into a single unit is known as Encapsulation.

For example: capsule, it is wrapped with different medicines.



- ✓ **In OOP, data and methods operating on that data are combined together to form a single unit, this is referred to as a Class.**
- ✓ Encapsulation is the mechanism that binds together code and the data it manipulates and keeps both safe from outside interference and misuse.
- ✓ **The insulation of the data from direct access by the program is called —data hiding.** Since the data stored in an object cannot be accessed directly, the data is safe i.e.,the data is unknown to other methods and objects.

4. Polymorphism:

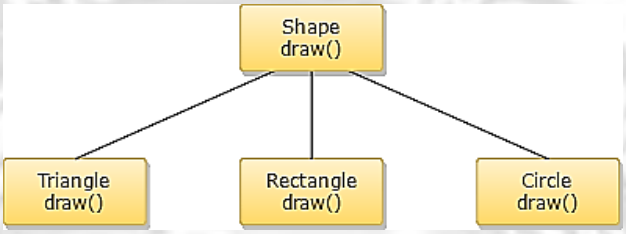
- ✓ Polymorphism is a concept by which we can perform a single action by different ways. It is the ability of an object to take more than one form.
- ✓ The word "poly" means many and "morphs" means forms. So polymorphism meansmany forms.
- ✓ An operation may exhibit different behaviors in different instances. The behavior depends on the data types used in the operation.
- ✓ For Example:- Suppose if you are in a classroom that time you behave like a student, when you are in the market at that time you behave like a customer, when you at your home at that time you behave like a son or daughter, Here one person present in different-different behaviors.

- Two types of polymorphism:

1. **Compile time polymorphism / Method Overloading: -** In this method, object is bound to the function call at the compile time itself.
2. **Runtime polymorphism / Method Overriding: -** In this method, object is bound to the function call only at the run time.

- **<u>In java, we use method overloading and method overriding to achieve polymorphism.</u>**
- **Example:**

1. draw(int x, int y, int z)
2. draw(int l, int b)
3. draw(int r)



5. Abstraction:

- ✓ Abstraction refers to the act of representing essential features without including the background details or explanations. **i.e., *Abstraction* means hiding lower-level details and exposing only the essential and relevant details to the users.**
- ✓ For Example: - Consider an ATM Machine; All are performing operations on the ATM machine like cash withdrawal, money transfer, retrieve mini-statement…etc. but we can't know internal details about ATM.
- ✓ Abstraction provides advantage of code reuse.
- ✓ Abstraction enables program open for extension.
- ✓ **In java, abstract classes and interfaces are used to achieve Abstraction.**

6. Inheritance:

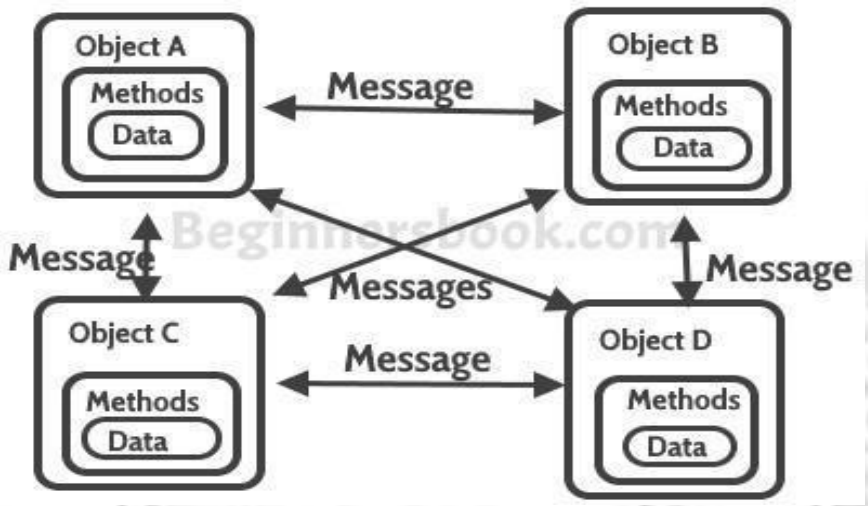
- ✓ **Inheritance in java** is a mechanism in which one object acquires all the propertiesand behaviors of another object.
- ✓ The idea behind inheritance in java is that we can create new classes that are built upon existing classes. When we inherit from an existing class, we can reuse methods and fields of parent class, and we can add new methods and fields also.
- ✓ Inheritance represents the **IS-A relationship**, also known **as *parent-child* relationship**.
- ✓ For example:- In a child and parent relationship, all the properties of a father areinherited by his son.
- ✓ **Syntax of Java Inheritance**

```
class Subclass-name extends Superclass-name
{
//methods and fields
}
```

7. Message Passing:

Message Communication:

- ✓ Objects interact and communicate with each other by sending ***messages*** to eachother. This information is passed along with the message as parameters.



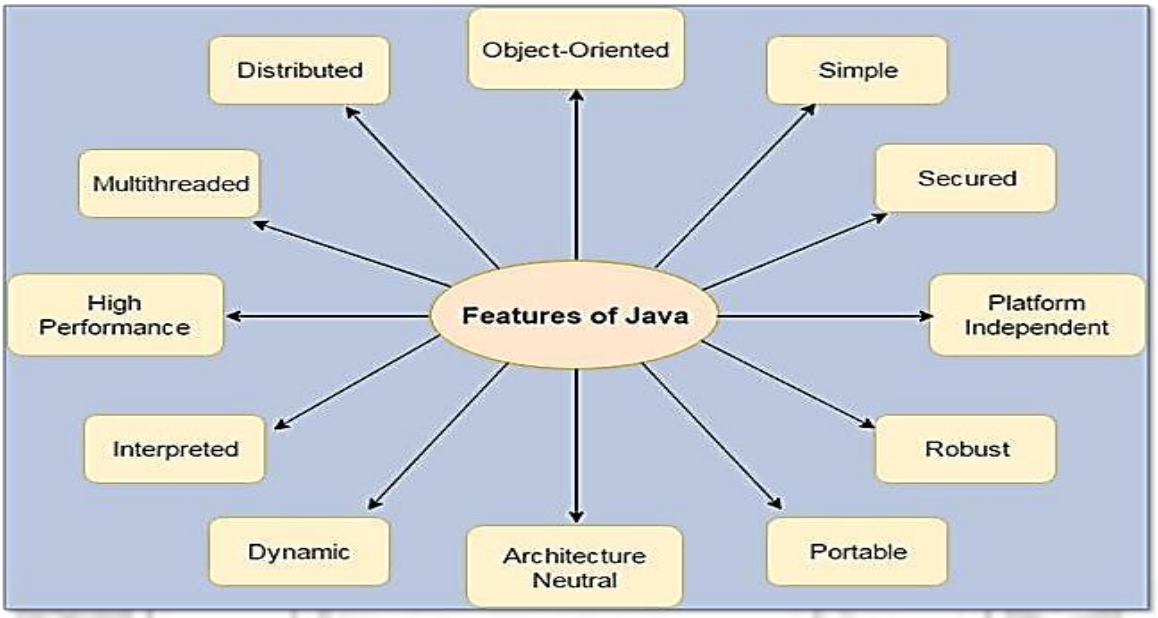
- ✓ A message for an object is a request for execution of a procedure and therefore will invoke a method (procedure) in the receiving object that generates the desired result.
- ✓ Message passing involves specifying the name of the object, the name of themethod (message) and the information to be sent.
- ✓ **Example:**

Employee.getName(name);
Where,
Employee – object name
getName – method name (message)
name - information

Java Buzzwords

The following are the features of the Java language:

1. Object Oriented
2. Simple
3. Secure
4. Platform Independent
5. Robust
6. Portable
7. Architecture Neutral
8. Dynamic
9. Interpreted
10. High Performance
11. Multithreaded
12. Distributed



1. Object Oriented:

- ✓ Java programming is pure object-oriented programming language. Like C++,Java provides most of the object oriented features.
- ✓ Though C++ is also an object oriented language, we can write programs in C++ without aclass but it is not possible to write a Java program without classes.
- ✓ Example: **Printing "Hello" Message.**

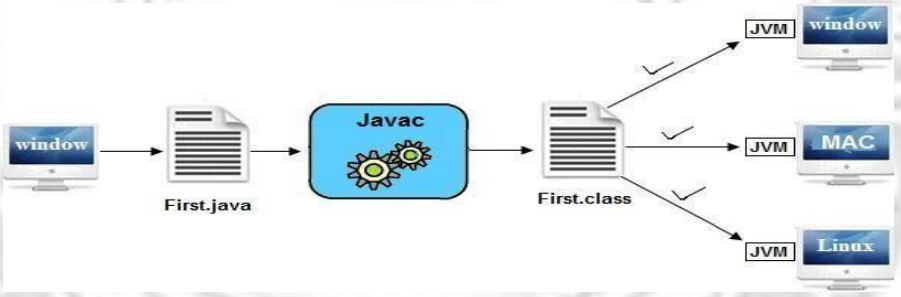
C++ (can be without class)	Java – No programs without classes and objects

With Class:
#include<iostream.h>
class display {
public:
void disp()
{
cout<<"Hello!";
}
};
main()
{
display d;
d.disp();
}
 	
Without class: #include<iostream.h>
void main()
{
clrscr();
cout<<"\n Hello!";
getch();
}
	With class:
	import java.io.*;
	class Hello {
	public static void main(String args[])
	{
	System.out.println("Hello!");
	}
	}
	
	Without class is not possible

2. **Simple:**
   - ✓ Java is Easy to write and more readable and eye catching.
   - ✓ Most of the concepts are drew from C++ thus making Java learning simpler.

3. **Secure :**
   - ✓ Since Java is intended to be used in networked/distributed environments, lot of emphasishas been placed on security.
   - ✓ Java provides a secure means of creating Internet applications and to access web applications.
   - ✓ Java enables the construction of secured, virus-free, tamper-free system.

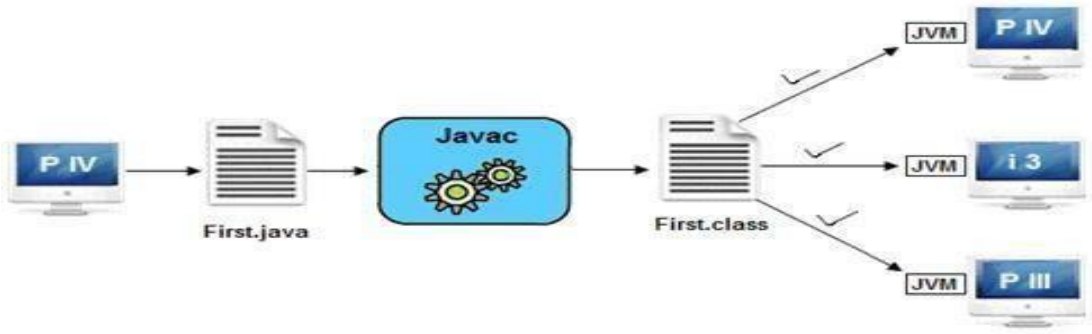
4. **Platform Independent:**
   - ✓ Unlike C, C++, when Java program is compiled, it is not compiled into platform-specific machine code, rather it is converted into platform independent code called **bytecode**.
   - ✓ The Java bytecodes are not specific to any processor. They can be executed in anycomputer without any error.
   - ✓ Because of the bytecode, Java is called as **Platform Independent.**



5. **Robust:**
   - ✓ Java encourages error-free programming by being strictly typed and performing run-time checks.

6. **Portable:**
   - ✓ Java bytecode can be distributed over the web and interpreted by **Java Virtual Machine (JVM)**
   - ✓ Java programs can run on any platform (Linux, Window, Mac)
   - ✓ Java programs can be transferred over world wide web (e.g applets)

7. **Architecture Neutral:**
   - ✓ Java is not tied to a specific machine or operating system architecture.
   - ✓ Machine Independent i.e Java is independent of hardware.
   - ✓ Bytecode instructions are designed to be both easy to interpret on any machine and easilytranslated into native machine code.



8. **Dynamic and Extensible:**
   - ✓ Java is a more dynamic language than C or C++. It was developed to adapt to an evolvingenvironment.
   - ✓ Java programs carry with them substantial amounts of run-time information that are usedto verify and resolve accesses to objects at run time.
9. **Interpreted:**
   - ✓ Java supports cross-platform code through the use of Java bytecode.
   - ✓ The Java interpreter can execute Java Bytecodes directly on any machine to which theinterpreter has been ported.

10. **High Performance:**
    - ✓ Bytecodes are highly optimized.
    - ✓ JVM can execute the bytecodes much faster.
    - ✓ With the use of Just-In-Time (JIT) compiler, it enables high performance.

11. **Multithreaded:**
    - ✓ Java provides integrated support for multithreaded programming.
    - ✓ Using multithreading capability, we can write programs that can do many tasks simultaneously.
    - ✓ The benefits of multithreading are better responsiveness and real-time behavior.

12. **Distributed:**
    - ✓ Java is designed for the distributed environment for the Internet because it handles TCP/IP protocols.
    - ✓ Java programs can be transmit and run over internet.