**Java** - a new object-oriented language that is receiving wide attention from both industry and

academia.

* developed by James Gosling and his team at Sun Microsystems in California.
* based on C and C++ and was originally intended for writing programs that control consumer appliances
* was first called Oak
* the name was already taken, so the team renamed it Java.
* often described as a Web programming language because of its use in writing programs called applets that run within a Web browser.
* platform independent
* class-based object-oriented programming

**Java Classes**

* contain instructions that tell a computer what to do as well as the source code using file using an editor.

**Java Source**

* like a word processing document.
* has instructions written in the java programming language.

**Editor**

* barebones word processor that doesn’t have all the fancy capabilities you typically find in a word processor.

**Classes and Objects**

**Object**

* a thing, both tangible and intangible

**Class**

* kind of mold or template that dictates what objects can and cannot do.

*An object is called an* ***instance*** *of a class. An object is*

*an instance of exactly one class. An instance of a class belongs to the class.*

**Attribute**

* characteristic of an object

**Behavior**

* action that an object can perform.

**Messages and Method**

**Message**

* To instruct a class or an object to perform a task

**Method**

* a class or an object to process the message it receives, it must possess a matching method

**Class method**

* method defined for a class

**Instance method**

* a method defined for an object

**Argument**

* value we pass to an object

**Main Principles of Object-Oriented Programming:**

**Abstraction (***Hides Complexity***)**

* means using simple things to represent complexity

**Encapsulation (***Data Security***)**

* describes the ability of an object to hide its data and methods from the rest of the world and is one of the fundamental principles of object-oriented programming.

**Inheritance (***Code reusability***)**

* an important feature of object-oriented programming languages.
* enables classes to include properties of other classes.

**Polymorphism (***Code reusability***)**

* the ability for different objects to respond differently to the same message.

**Concept of OOP in Java**

* OOP concepts allow us to create specific interactions between Java objects.

**Object oriented**

* Java can be easily extended since it is based on the Object model.

**Simplicity**

* Java is simpler than most other languages that are used to create server applications, because of its consistent enforcement of the object model.

**Portability**

* Java is portable across platforms. It is possible to write platform-dependent code in Java, and it is also simple to write programs that move seamlessly across systems.

**Strong typing**

* Before you use a field, you must declare the type of the field.

**Security**

* The design of Java bytecodes and JVM specification allow for built-in mechanisms to verify the security of Java binary code.

**2 core types of Java program**

**Java applications**

* the standalone programs written in Java to carry out certain tasks.
* It can be either a Command-Line Interface (CLI) or Graphical User Interface (GUI).

**Java Applets**

* the Java programs that are embedded inside HTML files and can be downloaded into a Java-capable browser
* you can do anything ranging from animations to complete games and utilities that can be executed over the Internet.

**JAVA DEVELOPMENT ENVIRONMENT**

**JVM** *(Java Virtual Machine)*

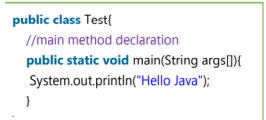
* is an abstract machine.
* it doesn't physically exist.
* available for many hardware and software platforms.
* JVM, JRE, and JDK are platform

**JRE *(****Java Runtime Environment)*

* also written as Java RTE.
* set of software tools which are used for developing Java applications.
* used to provide the runtime environment.

**JDK***(Java Development Kit)*

* a software development environment which is used to develop Java applications and applets.
* physically exists
* contains JRE + development tools
* an implementation of any one of the below given Java Platforms released by Oracle Corporation:



**STRUCTURE OF JAVA PROGRAM**

**Documentation Section**

* used to improve the readability of the program.
* consists of comments in Java which include basic information

There are three types of comments that Java supports

* Single line Comment
* Multi-line Comment
* Documentation Comment

**Package Statement**

* can be only one package statement in a Java program, and it must be at the beginning of the code before any class or interface declaration.

**Import Statement**

* used to refer to the classes stored in other packages.
* always written after the package statement, but it must be before any class declaration.

**Interface Section**

* is used to specify an interface in Java.
* is a lot like a class in Java, but it contains only constants and method declarations.

**Class Definition**

* defines the information about the user-defined classes in a program.

**Main Method Class**

* the most important section of your Java program.
* **public class Test:**

- creates a class called Test

- You should make sure that the class

name starts with a capital letter, and the public word means it is accessible from any other

classes.

* **Comment:**

**-** To improve the readability, we can use comments to define a specific note or

functionality of methods, etc for the programmer.

* **Braces**

-curly brackets are used to group all the commands together. To make sure

that the commands belong to a class or a method

* **public static void main**

- when declared public, means that it can be used outside of this class as well

- static means that we want to access a method without making its

objects.

- void indicates that it does not return any value

* **String[] args**

- an array where each element is a string

* **System.out.println()**

- used to print the output on the screen