Code at Random
(OPC) PVT. LTD.

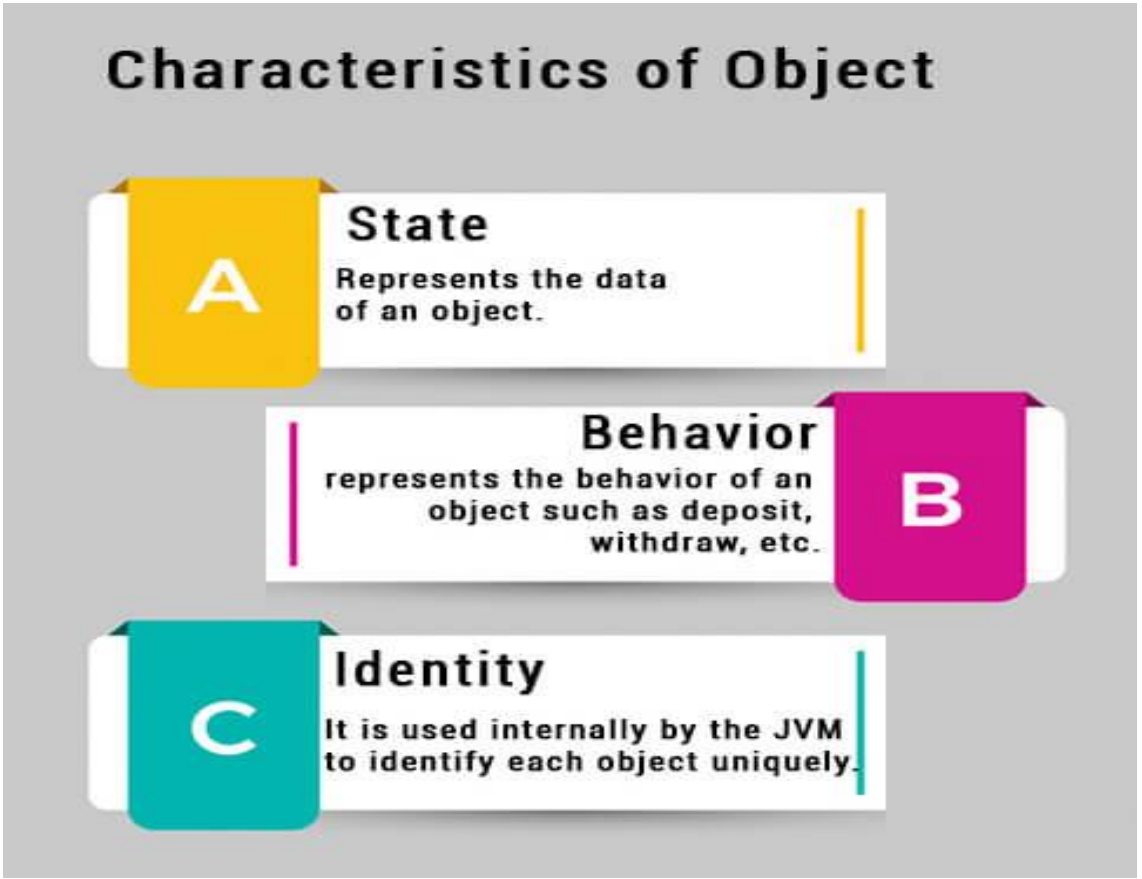
Elementary Concepts of Java

What Is An Object??

- An **object** is the name of any person, place, things or entity. Every object has the following inbuilt characteristics: **Identity, State and Behaviour**.

- **Identity:** It is the name by which an object is recognized. For example: TV, box, etc.
- **State:** The object may be in any state i.e. either processed or unprocessed. For example:
  1. ON state
  2. OFF state
  3. Out of order state.
- **Behaviour:** This refers that what an object is capable of doing.

Thus an object is an identifiable entity with some characteristics and behaviour of the object can be described by defining common states and behaviour of all the objects of a particular type in the form of a class.



Classes And Objects In JAVA

Classes and Objects are basic concepts of Object Oriented Programming which revolve around the real life entities.

CLASS

- A class is a user defined blueprint or prototype from which objects are created.
- It represents the set of properties or methods that are common to all objects of one type.
- Class declaration includes class keyword, class name, body of class surrounded by braces.
- A class is also considered as an **object factory** because the class is basically an object maker.
- It contains all the attributes (data members and member functions) to create an object and the statements that describe the operations that the object is going to perform.

Classes And Objects are interretable

- A class is the blueprint for an object, whereas an object is an instance of a class. Thus, classes and objects are interrelated.
- With one class, we can create several objects of similar kinds. Thus, a class is also referred to as an object factory.
- A Class is a structure of an object, meaning, all the necessary elements (data members + member functions) of the class are present in the object. That’s why it is known as instance of class.
- Objects are also termed as class tags or ***entities***.

Comments In Java

- Comments are text notes added to the program to provide explanatory information about the source code.
- They are used in a programming language to document the program and remind programmers of what tricky things they just did with the code and also helps the later generation for understanding and maintenance of code.
- The compiler considers these as non-executable statements.
- These are of two types: -
  - **Single-line Comments: -** The comments in single line are called single-line comments. These are written like: - **//comments**
  - **Multi-line Comments: -** The comments in multi-line are called multi-line comments. These are written like: -

    /* Comments

    Comments*/

Exceptions and its types

- When the program does not give the desired output, then we see it contains error. The unexpected situation which occurs during the execution of the program is called exception or error. There are three types of error:

a. **Compile time error**- The error which occurs during the compilation of the program, is called the Compile time error.
Eg: Syntax error, means the error which occurs when we violate the syntaxial rule of a programming language. i.e., missing semicolons, undeclared variable, mis-match curly braces, etc.

b. **Semantic error**- The error which occurs when we issue some meaningless statements is called semantic error. Its types are:

  - Logical Errors: The errors which occurs due to wrong implementation of logic is called logical errors. Eg: Infinite loop
  - Run-Time Errors: The error which occurs due to the execution of the program is called run-time errors.

Print Statements

In Java, there are two statements in Java to print something: -

- **System.out.print() –** This statement is used to print contents in same line.

For Example: - System.out.print(“Hello”);

System.out.print(“World”);

Output: - Hello World

- **System.out.println() –** This statement is used to print contents in different line.

For Example: - System.out.println(“Hello”);

System.out.println(“World”);

Output: - Hello

World

Syntax Of Java Program

Program (Source Code)

```
class MyClass {
  public static void main(String[] args) {
    System.out.println("Hello World");
  }
}
```

Note:

- **The curly braces {} marks the beginning and the end of a block of code.**
- **Each code statement must end with a semicolon.**

Explanation Of Code

- Every line of code that runs in Java must be inside a class.
- In our example, we named the class **MyClass**.
- Note: Java is case-sensitive: "MyClass" and "myclass" has different meaning.
- The name of the java file **must match** the class name.

- The main() method is required and you will see it in every Java program: public static void main(String[] args)
- Any code inside the main() method will be executed.
- For now, just remember that every Java program has a class name which must match the filename, and that every program must contain the main() method.

- Inside the main() method, we can use the println() method to print a line of text to the screen: **System.out.println("Hello World");**

A Sample Java Program

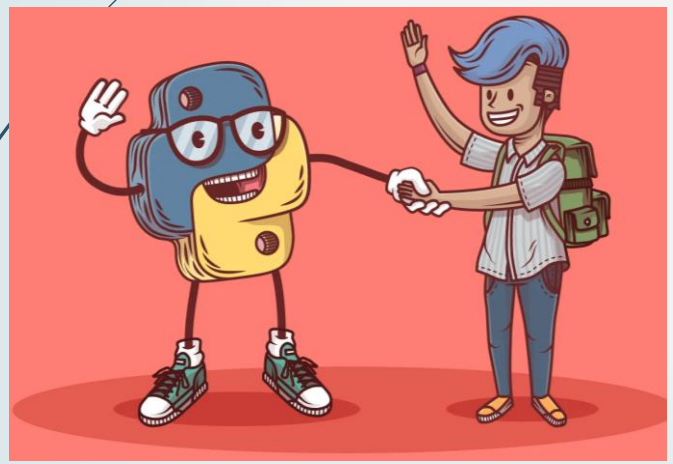
Program (Source Code)

```
public class hello_world // class declared
{
    public static void main(String Args[ ]) // Function prototype
    {                                        // Function starts

       System.out.print("Hello, ");    // Simple Print Statement
       System.out.print("My name is Aman"); //Semicolon is important
       System.out.println(“Hi ”); // Println Statement
      System.out.println(“ Code”);  // Semicolon terminates the line
    }                                        // Function stops

}                                            // Class Ends
```

Output

```
Hello, My name is Aman
  Hi
Code
```

Happy Learning!!



Code at Random
(OPC)
PVT. LTD.