**Classes in Java**

A **class in Java** is a fundamental building block of object-oriented programming (OOP) language. In other words, a class is the basic unit of OOP.

According to OOPs concept in Java, a class is the blueprint/template of an object. It contains the similar types of objects having the same states (properties) and behavior.

In other words, a class can also be defined as “a class is a group of objects which are common to all objects of one type”.

**Objects in Java**

An object is a basic unit of an object-oriented programming language. It is any real-world thing that has properties and actions.

In other words, an entity that has state and behavior is known as object in Java. Here, the state represents properties and behavior represents actions or functionality.

For example, book, pen, pencil, TV, fridge, washing machine, mobile phone, etc. Objects in Java consists of states or attributes (called data members) and behavior (called methods).

An object is an instance of a class. Each instance of an object holds its own relevant data.

**Characteristics of an object in Java**

An object in Java has three characteristics:

1. **State:** State represents properties or attributes of an object. It is represented by instance variable. The properties of an object are important because the outcome of functions depends on the properties.

2. **Behavior:** Behavior represents functionality or actions. It is represented by methods in Java.

3. **Identity:** Identity represents the unique name of an object. It differentiates one object from the other. The unique name of an object is used to identify the object.

Let’s take a real-world example to understand all these points clearly.

**Class Declaration in Java**

A class can be declared using the keyword class followed by a class name. It has also a body within braces. The general syntax to declare a class in Java is shown below:

Syntax:

modifierName class className

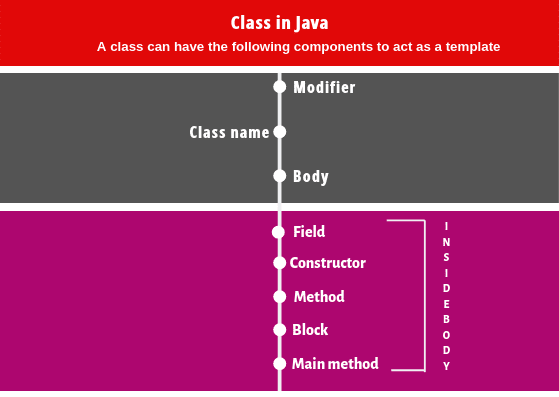
 {

   // class body.

 }

**Components of Class in Java**

In general, a class can have the following components to act as a template. It can be seen in the below figure and a brief explanation is given below.

[](https://www.scientecheasy.com/2020/07/java-classes-objects.html/)

1. **Modifiers:** A class can be either a public or default [access modifier](https://www.scientecheasy.com/2020/06/access-modifiers-in-java.html/). But the members of class can be public, private, default, and protected. All these are the access modifiers.

2. **Class name:** By convention, a class name should begin with a capital letter and subsequent characters lowercased (for example Student).

If a name consists of multiple words, the first letter of each word should be uppercased ( for example CollegerStudent).

A class name can also start with an underscore ”\_”. The following class name can be valid such as \_, \_Student. Keywords cannot be a valid class name. For example, class, true, null, etc are not accepted as a class name.

3. **Body:** Every class’s body is enclosed in a pair of left and right braces. In the body, a class can contain the following members.

class class\_name

{

     1. Field declarations;

    2. Constructor declarations;

    3. Method declarations;

    4. Instance block declarations;

    5. Static block declarations;

} Ends here.

If you are writing a simple program, everything must reside inside the body of the class. Let’s see in brief about all the members of the class.

**Fields:** Fields are data member [variables](https://www.scientecheasy.com/2020/05/variables-in-java.html/) of a class that stores data or value in it. It specifies the state or properties of the class and its object. It may be a local variable, instance variable, or [static variable](https://www.scientecheasy.com/2020/06/java-static-variable.html/).

**Constructor:** A [constructor](https://www.scientecheasy.com/2020/06/constructor-in-java.html/) is used to create an object. Every class must be at least one constructor otherwise, no object can be created of the class.

If you don’t explicitly define a constructor, the compiler automatically adds a default constructor inside the class. A constructor can be divided into two types, such as default constructor and user-defined constructor.

**Method:** A [method](https://www.scientecheasy.com/2020/06/java-methods.html/) defines an action or behavior of the class that a class’s object can perform. It has a body within braces. In the body, we write code that performs actions. It may be an instance method or a static method.

**Block:** A [block](https://www.scientecheasy.com/2020/06/instance-block-in-java.html/) is mostly used to change the default values of variables. It may be an instance block or static block.

**Interface:** An interface in Java is a mechanism that is used to achieve complete abstraction. It is basically a kind of class but can have only abstract method declarations and constants as members.

It is used to achieve multiple inheritances in Java. You will learn more details about the interface in the further tutorial.

**Method main:** A class has also the [main method](https://www.scientecheasy.com/2020/06/main-method-in-java.html/) that provides the entry point to start the execution of any program. The main method is static in nature. The signature of the main method is as follows:

public static void main(String[] args)

{

// This is a static region.

.............

}

The JVM executes everything between the curly braces { } of the main method. Every java program has at least one class and at least one main method.

**Flow of Execution of Program in Java**

1. Out of five elements declared inside the class body, the static variable, static block, and static method are executed first during the loading of the dot class file.

2. The instance variable is executed during object creation.

3. When the object is created, first, an instance block is executed before the execution of a constructor.

4. After the execution of an instance block, the constructor part will be executed.

5. After the execution of the constructor part, the instance method is executed.

6. The local variable is executed inside the method, constructor, or block.

## Difference between Class and Object in Java

There are the following differences between class and object in java. They are as follows:

1. A class is a user-defined data type whereas an object is an instance of class data type.

2. A class generates objects whereas an object gives life to a class.

3. Classes do not occupy memory location but objects occupy memory location.

4. Classes cannot be manipulated due to not available in the memory location but objects can be manipulated.

### Can we call Class as Data type in Java?

Yes, a class is also considered a user-defined data type. This is because a user creates a class.

**Key Points concerning Classes and objects in Java**

1. Objects, classes, data abstraction, encapsulation, inheritance, method overriding, polymorphism, and message passing are the fundamental terms and concepts to understand the concepts of object-oriented programming approach.

2. Objects are the basic runtime entities in object-oriented systems.

3. All objects in a system take a separate memory space independently of each other.

4. A class in OOPs represents a group of similar objects.

5. Once a class has been created, any number of objects belonging to that class can be created.

6. An object of a class is also known as an instance.

7. A class never represents an object, rather it represents data and actions that an object will have.

8. Members of a class consist of variables (data members) and methods (member functions).