<Codemithra />™

Explore | Expand | Enrich

<Codemithra />™

Class & Object

WHAT YOU'LL LEARN

- Class
- Object
- Life Cycle of an Object in Java
- Anonymous Object in Java
- Types of Classes in Java

CLASS AND OBJECT

ETHNUS
Explore | Expand | Enrich

Class

- In the object-oriented programming language, we design and develop an application program using classes and objects.OOPs Concepts in Java
- The underlying structure of all java programs is made up of classes.
- Anything we want to write in a java program must be encapsulated in a class.
- A class is a model to create objects. It means that we write properties and actions of objects in the class.

CLASS AND OBJECT

Explore | Expand | Enrich

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.

Syntax:

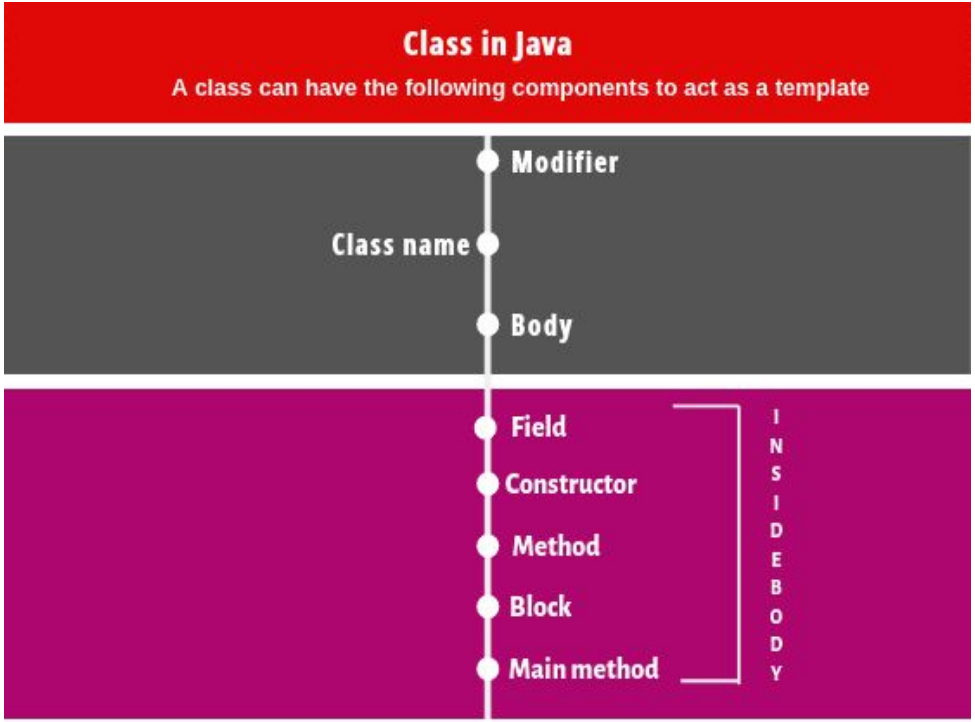
```
modifierName class className
{
// class body.
}
```

CLASS AND OBJECT

ETHNUS
Explore | Expand | Enrich

Class Declaration in Java

```
class class_name
{
 // Class body starts here.
 // Members of a class.
      1. Field declarations;
      2. Constructor declarations;
      3. Method declarations;
      4. Instance block declarations;
      5. Static block declarations;
 } Ends here
```



CLASS AND OBJECT

ETHNUS
Explore | Expand | Enrich

Main method in JAVA

A class has also the main method that provides the entry point to start the execution of program.

```
public static void main(String[] args)
{
 //  This is a static region.
      .............
}
```

CLASS AND OBJECT

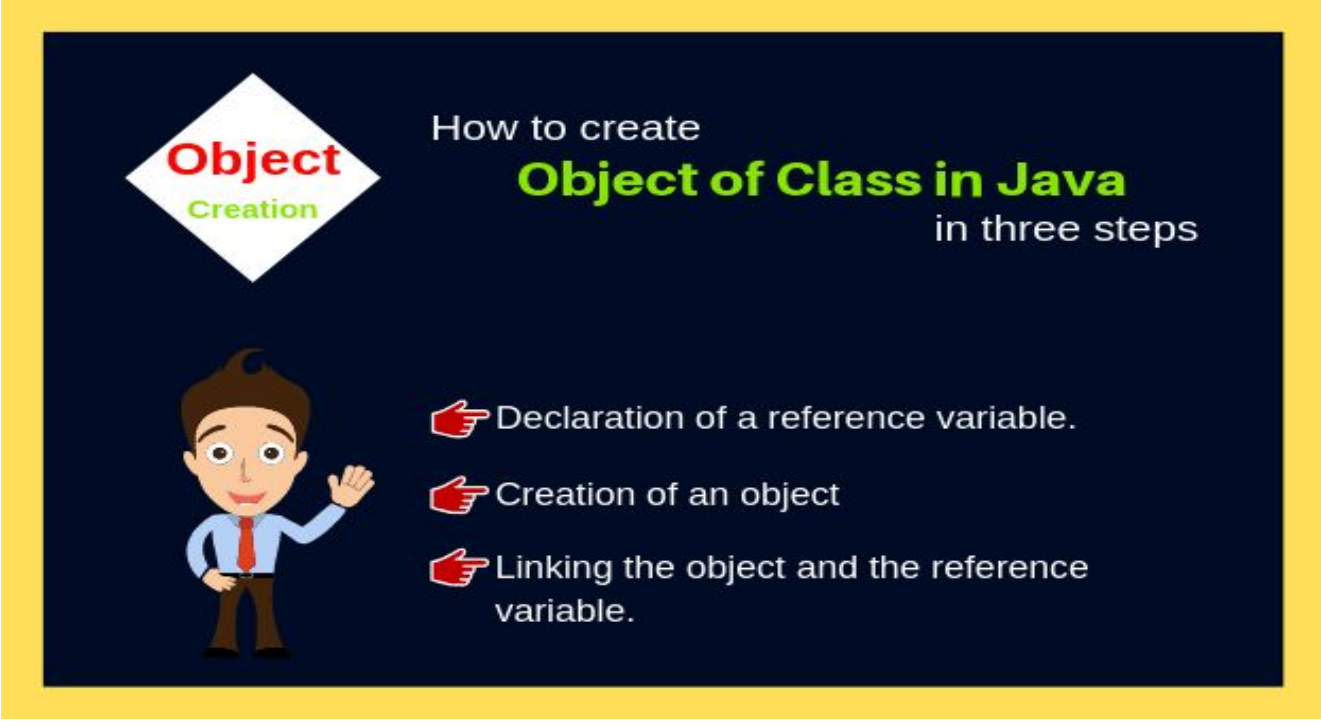
Object

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

CLASS AND OBJECT

Explore | Expand | Enrich

Object



CLASS AND OBJECT

ETHNUS
Explore | Expand | Enrich

Object

Declaration of Reference variable

```
Classname object_reference_variable; // Creating a reference variable of type Class.
```

Creating Object in Java

```
new Classname(); // Creating an object of class.
```

Linking Object and Reference variable

```
Classname object_reference_variable = new Classname();
```

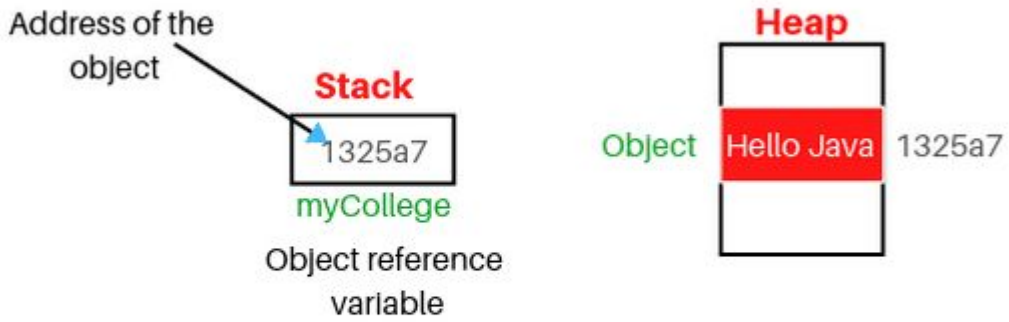
CLASS AND OBJECT

Explore | Expand | Enrich

Object

Memory Allocation of storing an object in Java

```
College myCollege=new College();
```



myCollege stores the address of the object on the stack memory.

new keyword allocates memory to store the object on the heap memory.

Fig: Memory allocation of storing an object in Java

CLASS AND OBJECT

Life cycle of an Object in Java

Life cycle of an object in Java

There are seven steps comes in the life cycle of object in java.

Step 1: Creation of .class file on disk

Step 2: Loading .class file into memory

Step 3: Looking for initialized static members of class

Step 4: Ways to initialize class in java

Step 5: Allocation of memory for object and reference variable

Step 6: Calling of the constructor of class

Step 7: Removing of object and reference variable from memory

Anonymous Object in Java

An object which has no reference variable is called anonymous object in Java.

Syntax:

```
new Class_name();
```

CLASS AND OBJECT

Anonymous Object in Java

```
package anonymousObject;
 public class Calculation
{
    int a;
    Calculation(int p)
    {
     a = p;
    }
   void area()
   {
     int area = a * a;
     System.out.println("Area of square: " +area);
   }
void perimeter(int b) {
int peri = 4 * b;
   System.out.println("Perimeter of square: " +peri);
 }
public static void main(String[] args)
 {
// Create anonymous object.
     new Calculation(50).area();
     new Calculation(10).perimeter(100);
     new Calculation(20).area();
     new Calculation(30).perimeter(200);
  } }
```

CLASS AND OBJECT

Types of class in Java

- Concrete Class
- Static Class
- Abstract Class
- Final Class
- Inner Class
- Public Class
- Private Class
- Singleton Class

CLASS AND OBJECT

Concrete Class

```
public class Concrete { // Concrete class
int a;
int b;
Concrete() {
 a = 50;
 b = 50;
}
void display() {
 int sum = a + b;
 System.out.println("Sum of two numbers: " +sum );}
public static void main(String[] args) {
Concrete c = new Concrete();
 c.display();}}
```

CLASS AND OBJECT

Explore | Expand | Enrich

Static Class

```
public class A
{
 static class B  {
  public void m1()
  {
     System.out.println("Static nested class method");
  }
 } public static void main(String[] args)
{
    B b = new B();
    b.m1();
   }
}
```

CLASS AND OBJECT

Abstract Class

```
public abstract class Hello {
 public void msg1() {
    System.out.println("msg1-Hello");
  }
abstract public void msg2();
}
public class Test extends Hello  {
  public void msg2()  {
   System.out.println("msg2-Test");
  }
public static void main(String[] args) {
  Test obj = new Test();
    obj.msg1();
    obj.msg2(); }}
```

CLASS AND OBJECT

Explore | Expand | Enrich

Final Class

```
final class Hello  {
 public void msg()  {
    System.out.println("msg-Hello");
  } }
public class DerivedClass extends Hello  {
public void msg()
{
 System.out.println("msg-DerivedClass");
 }}
public class Test  {
public static void main(String[] args)  {
 DerivedClass d = new DerivedClass();
  d.msg();}
```

CLASS AND OBJECT

Predefined Classes in Java

- Object Class
- Math Class
- String Class
- System Class
- Random Class
- Scanner Class
- Wrapper Class

/ethnuscodemithra

Ethnus Codemithra

/ethnus

/code_mithra

<Codemithra />™

https://learn.codemithra.com

ETHNUS™

Explore | Expand | Enrich

codemithra@ethnus.com

+91 7815 095 095

+91 9019 921 340