<https://java2blog.com/introduction-java-programming/>

<https://java2blog.com/java-threadpoolexecutor-example/>

**Introduction to Java Programming**

Java is one of the most used programming languages in the world for developing desktop application and web application. It has completed 22 years in the programming world.

## History of Java:

Java programming language was developed by James Ghosling, Patrick Naughton, Mike Sheridan at Sun Microsystems Inc. in 1991.

The initial name was **Oak** but it was renamed to **Java** in 1995 as OAK was a registered trademark of another Tech company.

### Evolution of Java:

There are many java versions that have been released. The current stable release of Java is Java SE 8 and the development of Java SE 9 is in progress and will be released soon.

* JDK Alpha and Beta (1995)
* JDK 1.0 (23rd Jan, 1996)
* JDK 1.1 (19th Feb, 1997)
* J2SE 1.2 (8th Dec, 1998)
* J2SE 1.3 (8th May, 2000)
* J2SE 1.4 (6th Feb, 2002)
* J2SE 5.0 (30th Sep, 2004): This version of Java have introduced major changes in java programming world. Some of them are:
  + [**Java Concurrency API**](https://www.java2blog.com/2016/12/java-threadpoolexecutor-example.html)
  + Var argument methods
  + Generics
  + Annotations
  + [**For each loop**](https://www.java2blog.com/2016/09/java-8-foreach-examples.html)
  + Enumerations
  + Static import
  + Autoboxing and unboxing etc.
* Java SE 6 (11th Dec, 2006)
* Java SE 7 (28th July, 2011) : Java 7 has following new major changes:
  + Multi-catch block
  + Support of String in Switch case
  + Try with resources etc.
* Java SE 8 (18th March, 2014) : Java 8 has introduced major programming paradigms. Some of them are:
  + [**Lambda Expressions.**](https://www.java2blog.com/2014/06/lambda-expressions-in-java-8.html)
  + [**Functional Interfaces**](https://www.java2blog.com/2016/09/java-8-functional-interface-example.html)
  + [**Default methods in**](https://www.java2blog.com/2014/06/interface-default-methods-in-java-8.html)interface.
  + [**Streams**](https://www.java2blog.com/2016/09/java-8-stream-filter-examples.html)
  + Enhanced Security etc

## Features of Java:

* **Simple:**  Java is easy to learn. It does not include concepts of pointers and operator overloading that were available in C++.
* **Object Oriented:** You can model everything into an [**object**](https://www.java2blog.com/2017/05/object-class-java.html) which has data and behavior. Java has incorporated various [**object-oriented concepts**](https://www.java2blog.com/2017/03/oops-interview-questions-answers-java.html)such as [**Abstraction**](https://www.java2blog.com/2017/04/abstraction-java-example.html), [**Encapsulation**](https://www.java2blog.com/2017/05/encapsulation-java-example.html), [**Polymorphism**](https://www.java2blog.com/2017/05/polymorphism-java-example.html), and inheritance.
* **Platform independence:**  Java is write-once, run-anywhere language. Once you compile java code, you can run it on any machine. For example: If you compile your code on Windows OS, you can easily run the same bytecode in linux.
* **Portable:** Java byte can be portable to any platform and can be executed on any platform.
* **Multi-threading:** Java has provided [**multithreading**](https://www.java2blog.com/2015/06/java-thread-example.html) feature which will help you to execute various task in parallel. It has introduced executor framework which is more robust and easy to use.
* **High performance:** Java can not be as fast as C and C++ as it is interpreted language but it has provided various features such as Just in time compiler to increase performance

|  |
| --- |
| Line number 4: public static void main(String[] args) { |

This is called main method in java. This is entry point for this program.

**public:** This is access modifier which is used to define visibility. Here main method will be visible to all other classes.

**static:** As this will be executed by JVM, it should be static.There should be no need to create object of this call.

**void:** This main method will not return anything.

**main:** “main” is method name here.

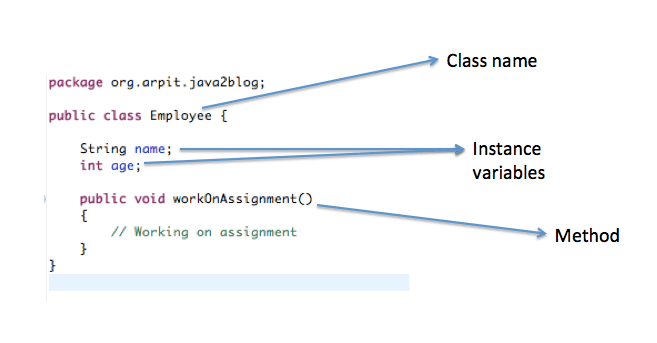
**String args[]:** you can pass arguments to java program using args[] array.

# Object and Class in java

Object: An entity that has state and behavior may be termed as Object.  
For example: Employee has a state with name, age, and department, and behavior such as working on the assignment.

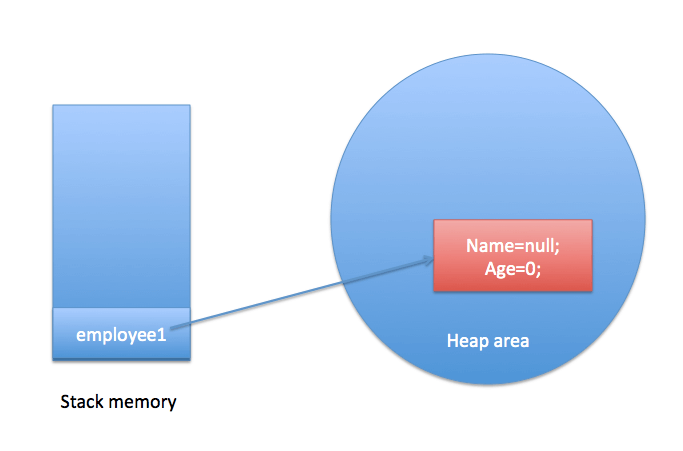
Class: A class is a blueprint/template that defines state and behavior of objects.  
For example: Employee is a class in above example.

Object**is an instance of**class**:** As object are created on the basis of blueprint/template that class provides, Object is an instance of the class.

You can create object of above class as below:

Employee employee1=new Employee();

Objects are stored in heap memory.  
When above statement gets called, physical object is created in the memory as below:



Employee employee1=new Employee();

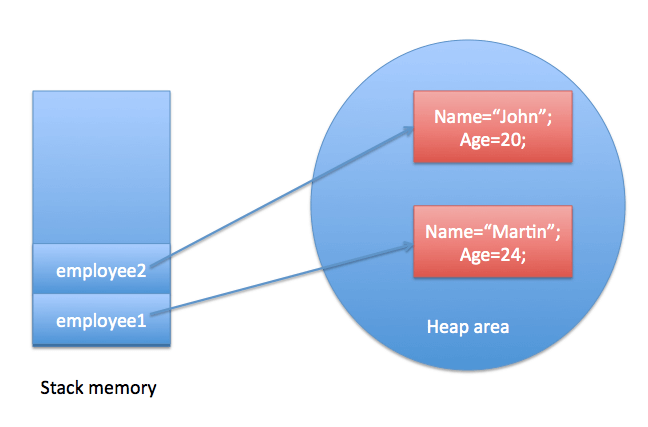
employee1.setName("Martin");

employee1.setAge(24);

Employee employee2=new Employee();

employee2.setName("John");

employee2.setAge(20);



Object is created using the [**constructor**](https://java2blog.com/constructor-java/) with the help of a new operator.

# Data types in java

Data types in java  refer to type of data that can be stored in variable. As Java is strongly typed language, you need to define datatype of variable to use it and you can not assign incompatible datatype otherwise the compiler will give you an error.

int d=”Hello”

The compiler will give you an error with this message – “Type mismatch: cannot convert from String to int”.  
There are two types of data types in java.

* Primitive data types
* Referenced data types.

### Assigning int to double(Widening)

Here we will assign int to double. As double takes more memory than int. This is widening operation.

public class DataTypesDemo {

public static void main(String[] args) {

int a=30;

double b=a;

System.out.println(a);

System.out.println(b);

}

}

When you run above program, you will get below output:

30  
30.0

### Assigning double to int(Narrowing or typecasting)

Here we will assign double to int. As double takes more memory than int. This is Narrowing operation.

public class DataTypesDemo {

public static void main(String[] args) {

double a=30.0;

int b=(int) a;

System.out.println(a);

System.out.println(b);

}

}

When you run above program, you will get below output:

30.0  
30

# Super keyword in java with example

Super Keyword in java is used to refer to object of the immediate superclass.Super keyword is used in the context of inheritance.  
Super keyword in java can be used at three level

* Variable level
* Constructor level
* Method level