



Introduction to Java

Objectives



Objectives

After completing this module, you should be able to:

- Understand the advantages of Java
- Understand where Java is used
- Understand how Memory Management is handled in Java
- Create a Java project in Eclipse and execute it
- Implement if - else construct in Java
- Develop Codes using various Data Types in Java
- Implement various Loops

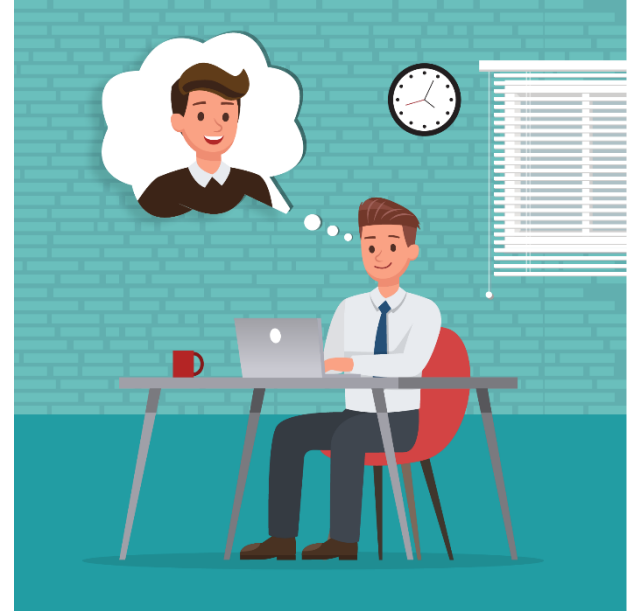




Introduction to Java

Meet John

John is a student of Computer Science in California University.



John dreams of becoming a “Programmer” like his elder brother.

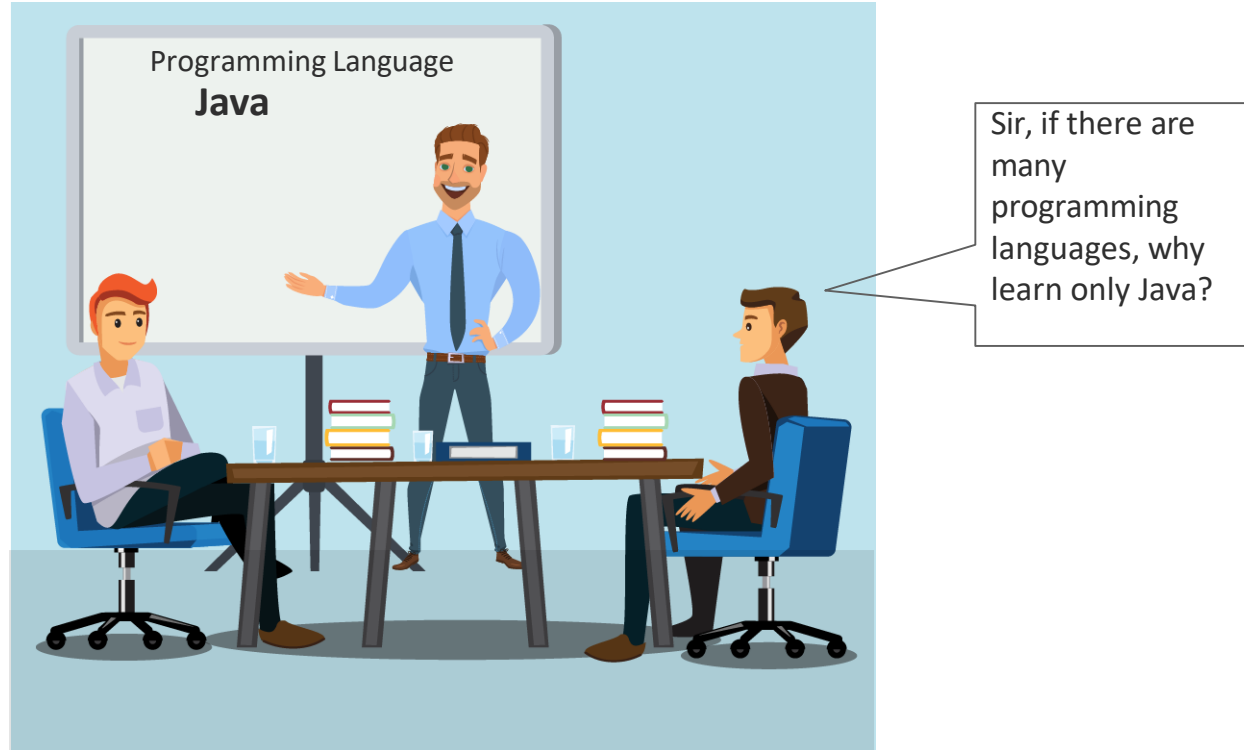
John Learns about Programming Languages

- One day in a class!!

There are many programming languages. Java is one of them. It's very important, we will learn about it now.

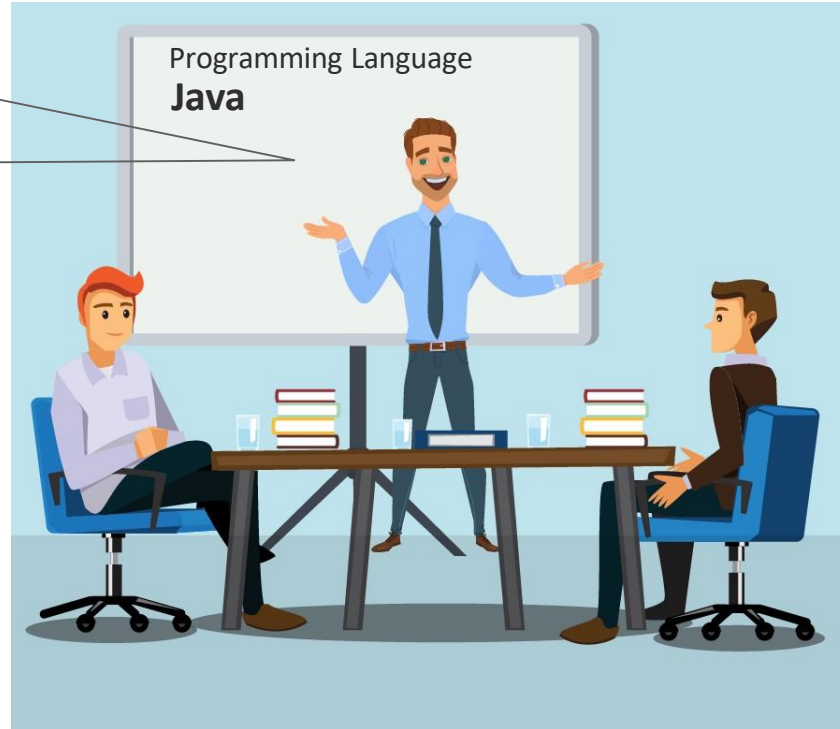


John is Confused!

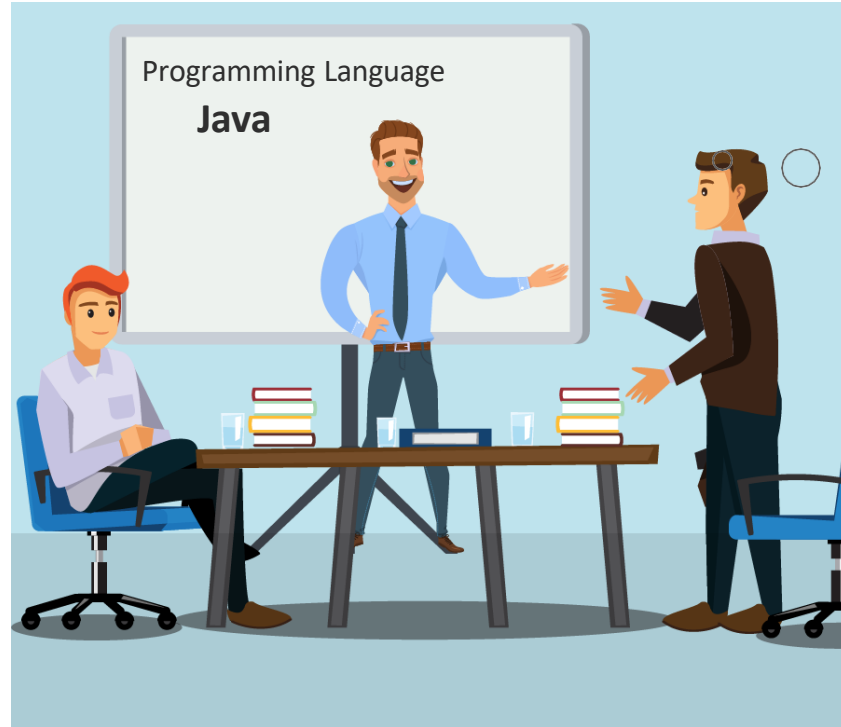


John gets a Bookish Answer

Because Java is highly secured, object oriented and most widely used language in the Industry.



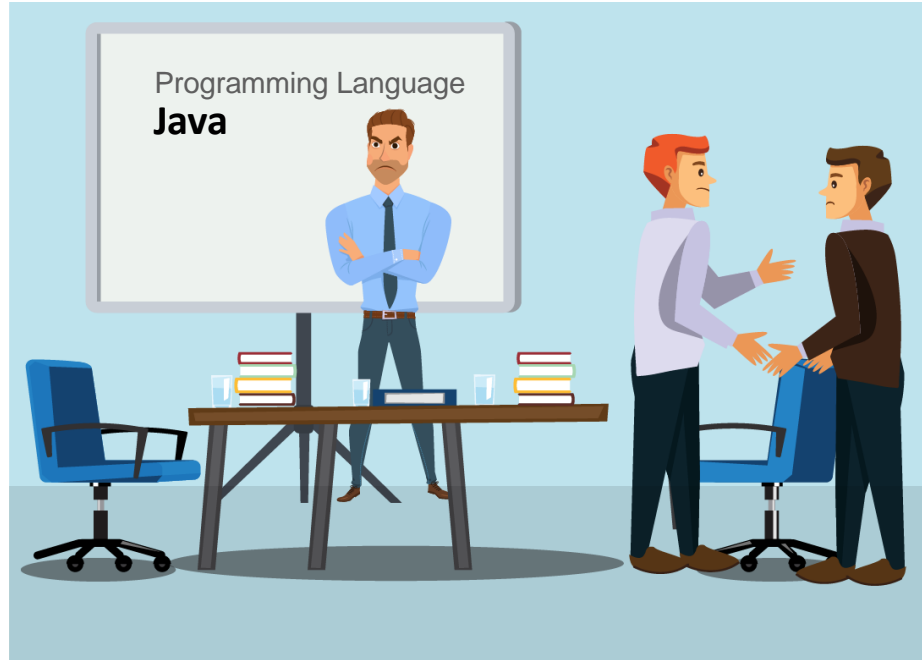
John is still Confused!



But where
do we use
it? Why
should I
learn Java?

John Heads Home with a Doubt

John was not able to relate anything on what was taught in the class



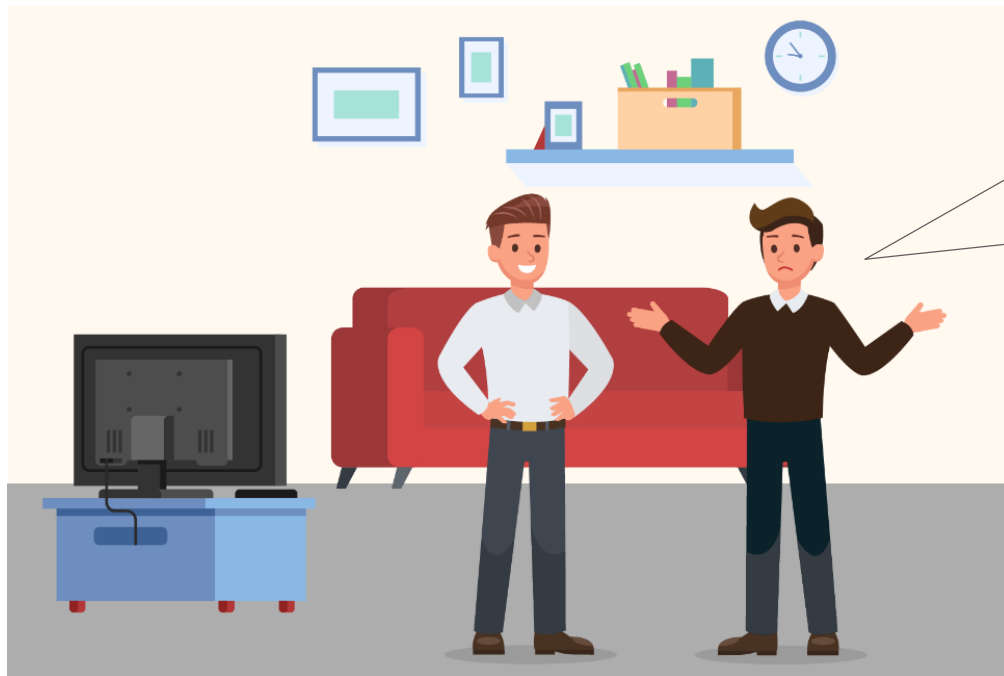
John meets his Brother

On returning home, John met his elder brother. John's brother realized that John was upset.

Hello John! How was your day?

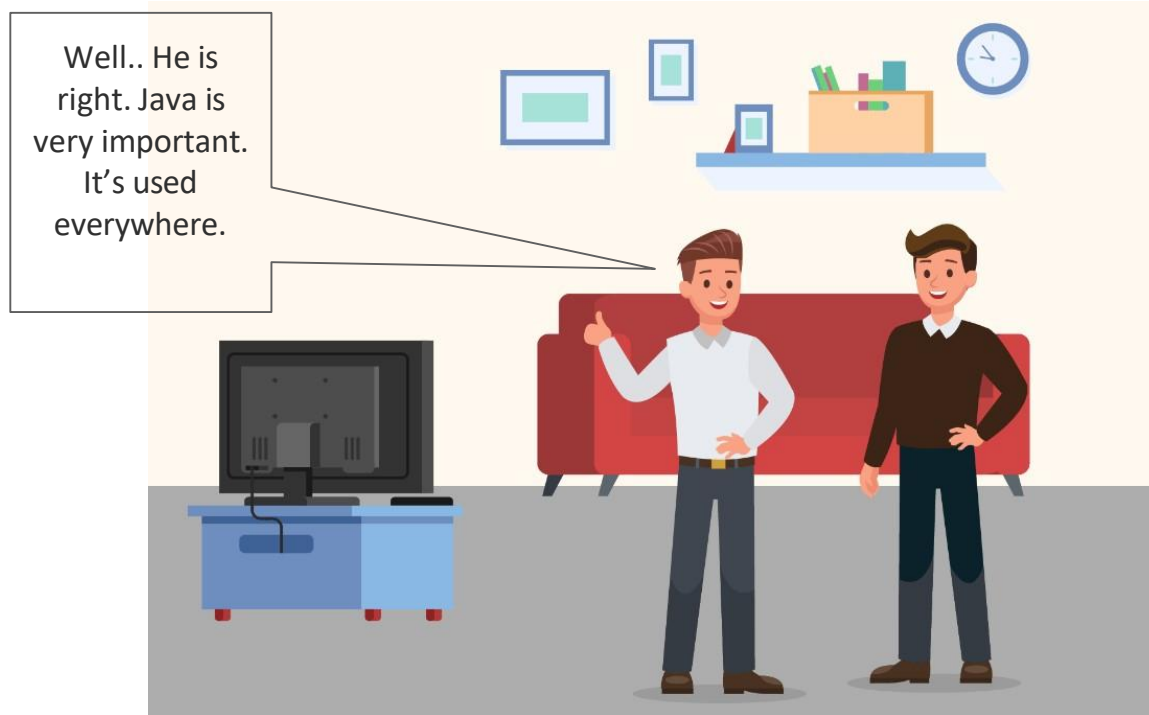


John Expresses his Concerns



My teacher started programming languages today. He started with Java and said it's very useful and important. I am not able to understand why?

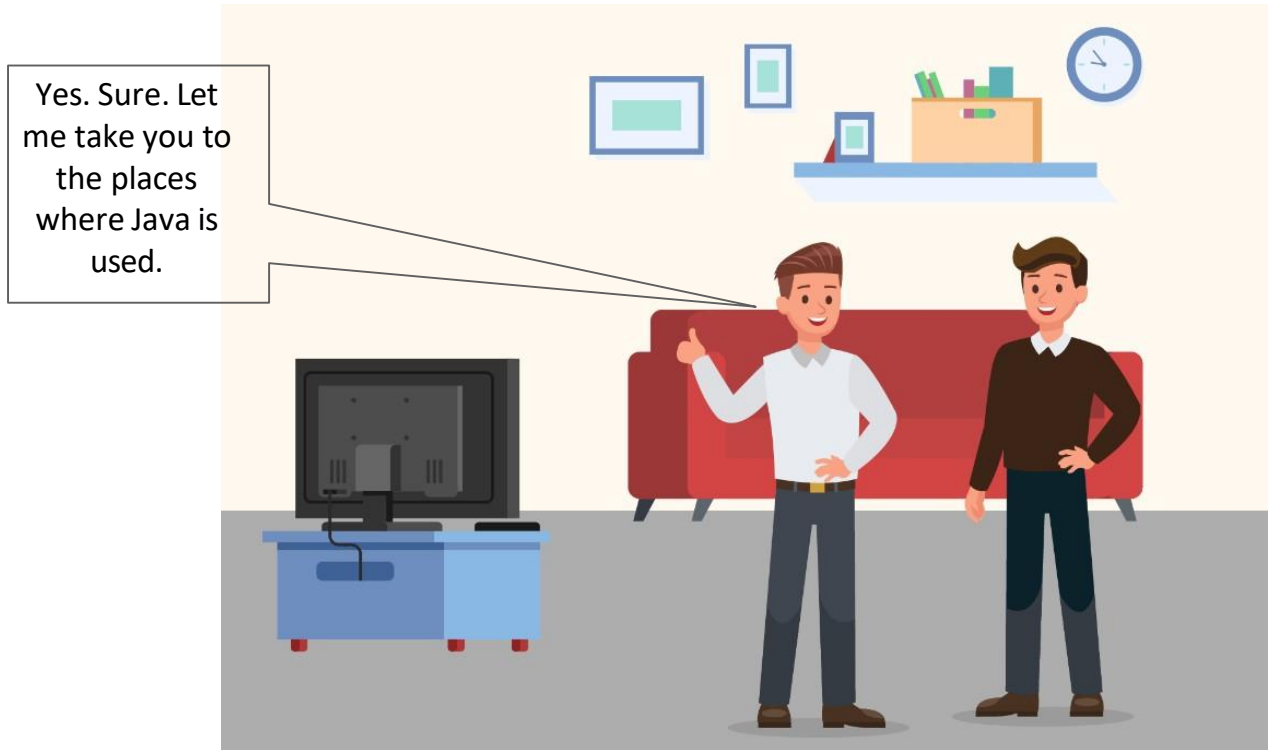
John gets an Answer!



John wants to know more..

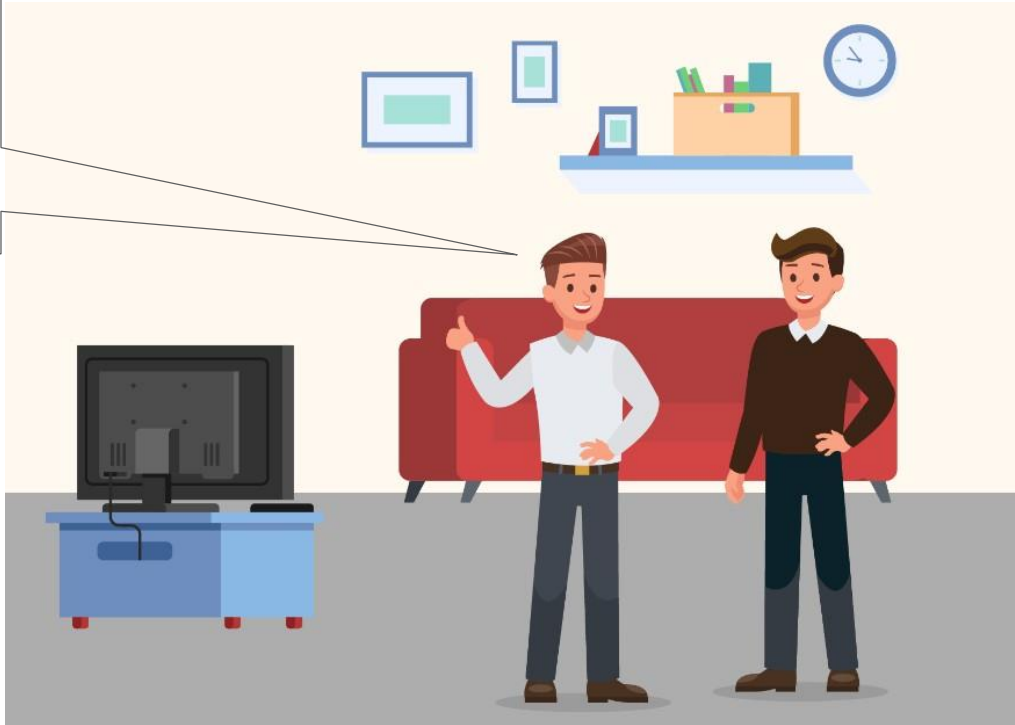


John gets Help!



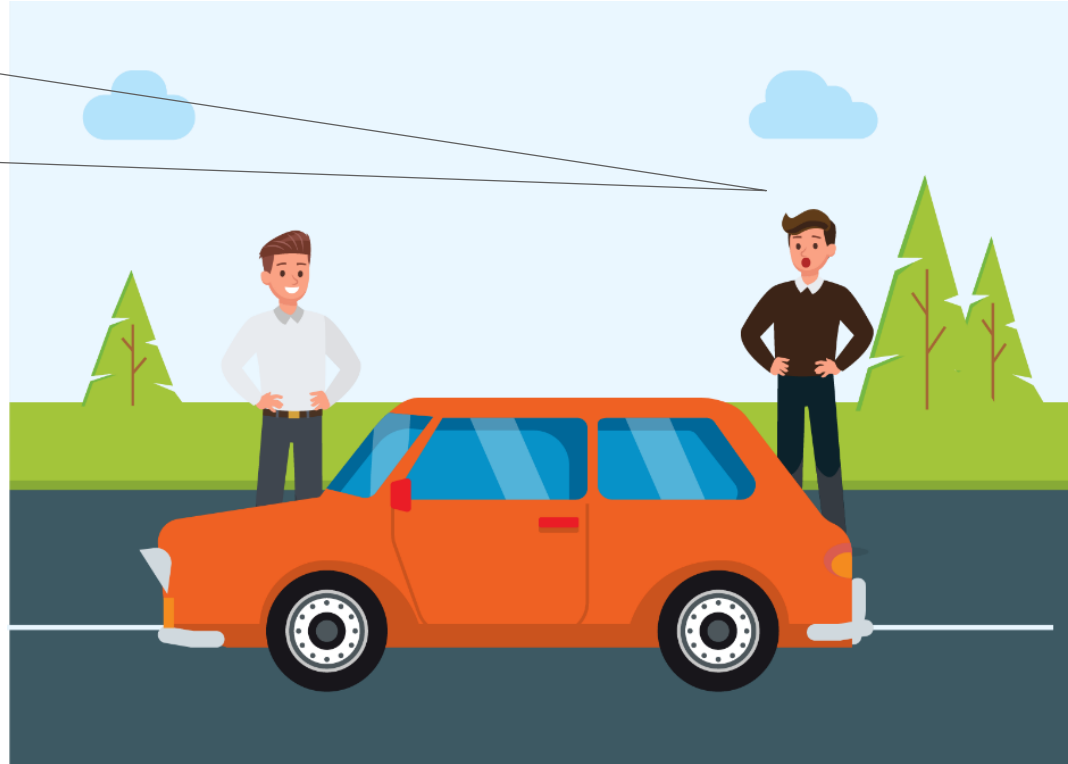
Uses of Java - 1

Look there. Do you see the set-top box near the TV? It works on Java..



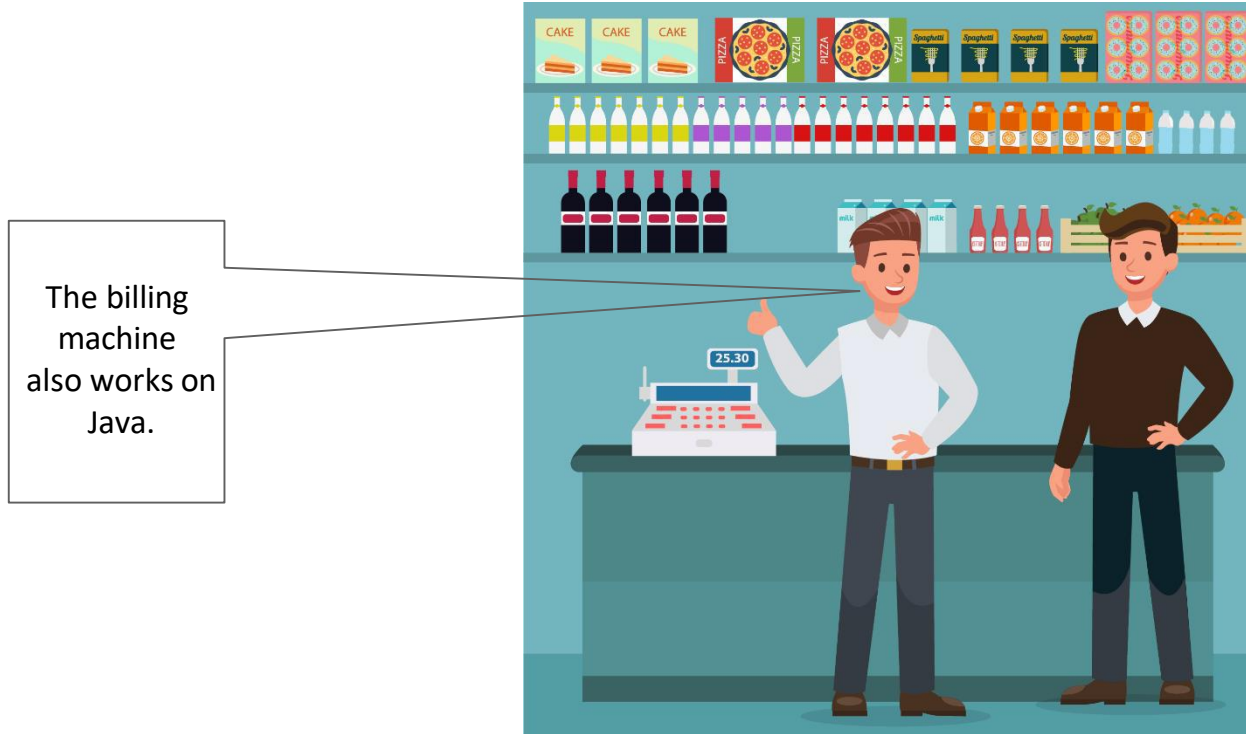
Uses of Java - 2

Our car uses GPS system to guide us and it's a very useful navigation tool. It works on Java.
Let's see some more places.



Uses of Java - 3

John's brother took him to the groceries shop.



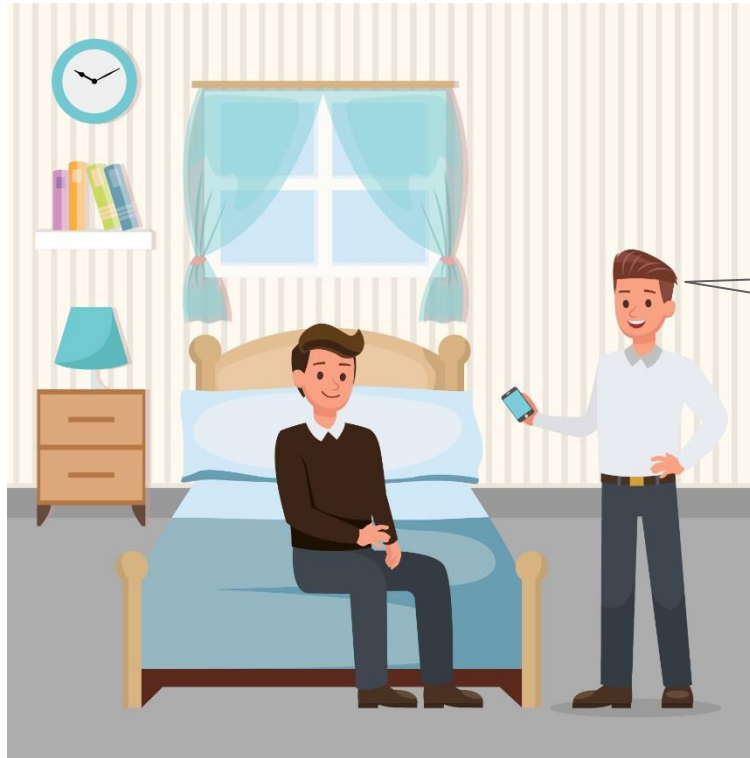
John is Convinced to Learn Java

While returning, John had understood that Java is almost everywhere.

So John, now that you know, Java is used in so many fields, you should start learning it if you wish to be a good programmer.



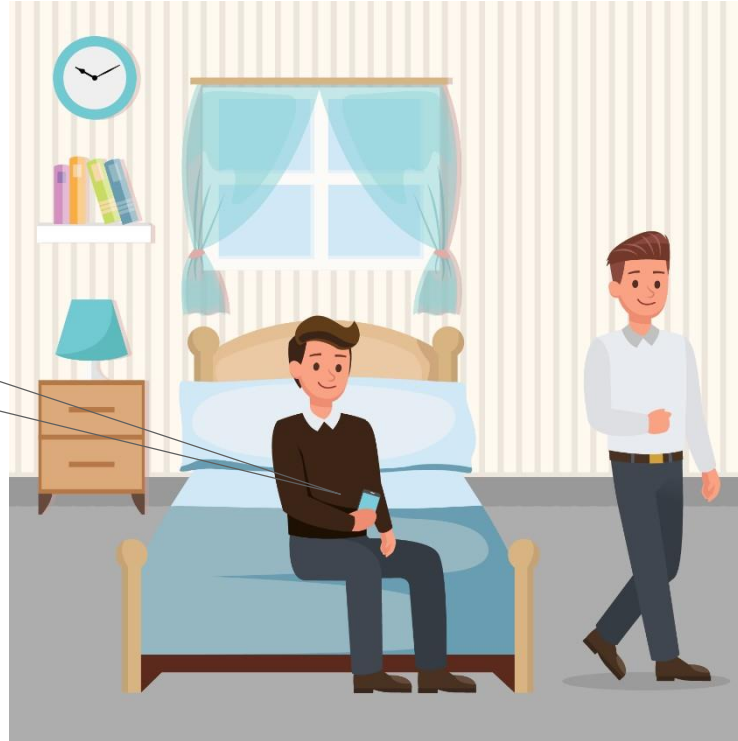
A Gift for John



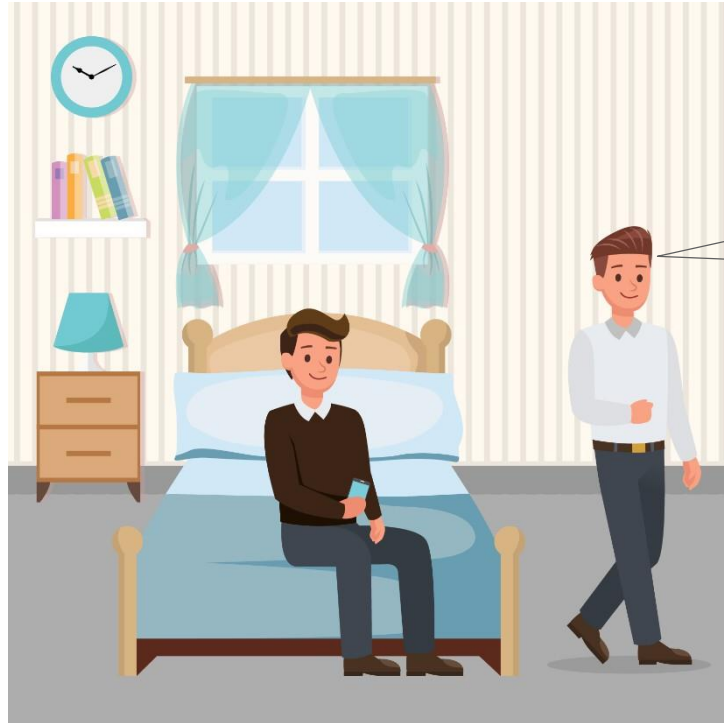
Look John. I
have bought
you a gift.

An Android Phone

Wow! An
android
phone!!
Thank you



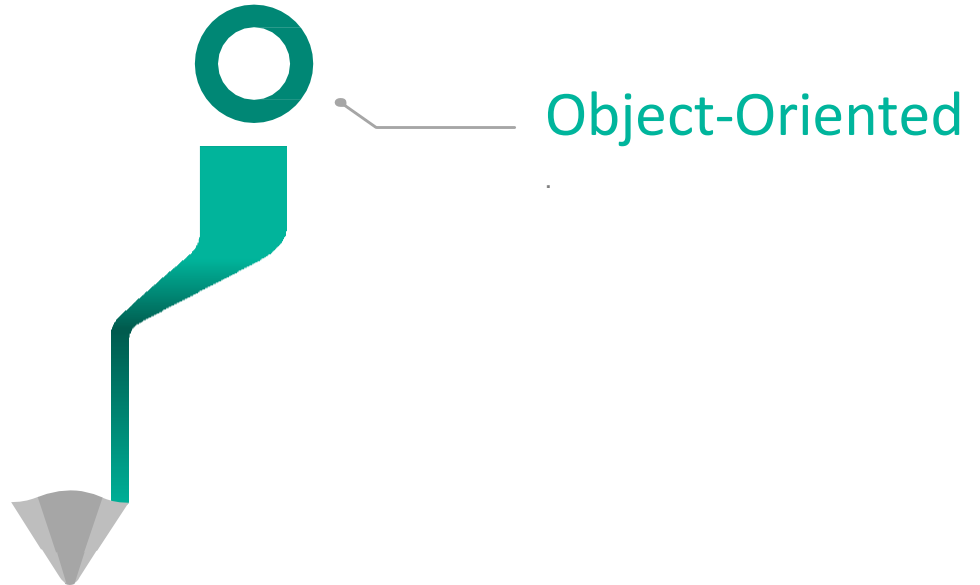
John is all set to Learn Java



Well. That works
on Java too..

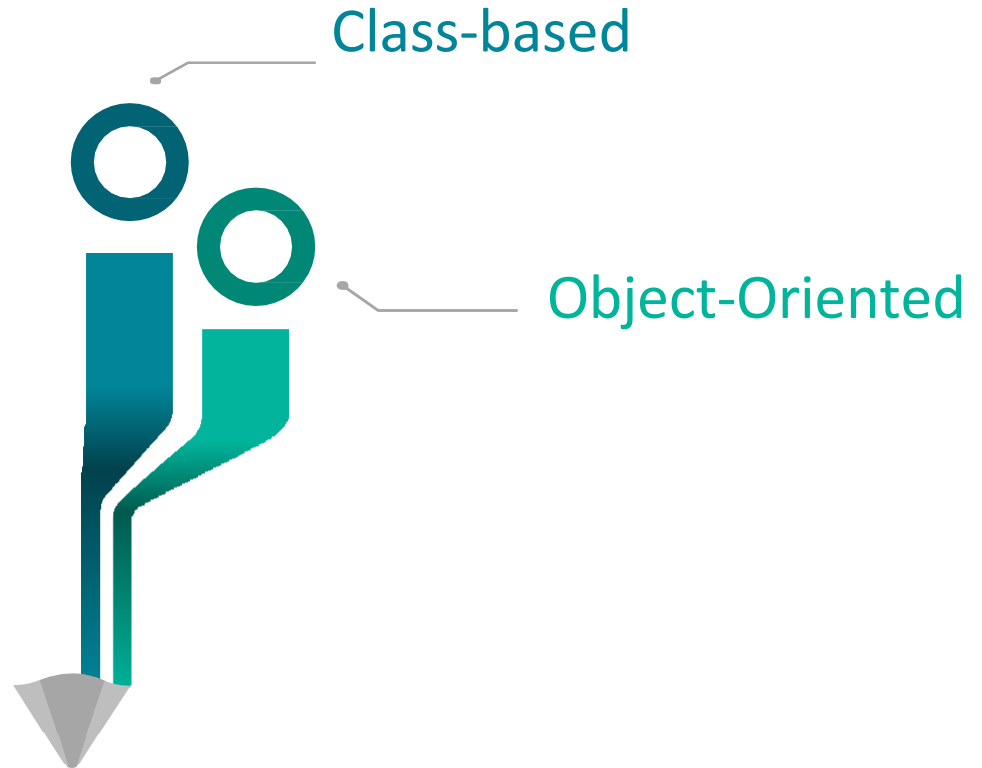
What is Java?

Java is a computer **programming language** that is -



What is Java?

Java is a computer **programming language** that is -



What is Java?

Java is a computer programming language that is -

Concurrent

Class-based

Object-Oriented



What is Java?

Java is a computer **programming language** that is -



Where is Java Used?



3 billion devices use Java Runtime Environment

To develop Android Applications

To develop Hadoop Applications

To develop business process management (BPM) tools

To develop web servers

To develop application servers etc.

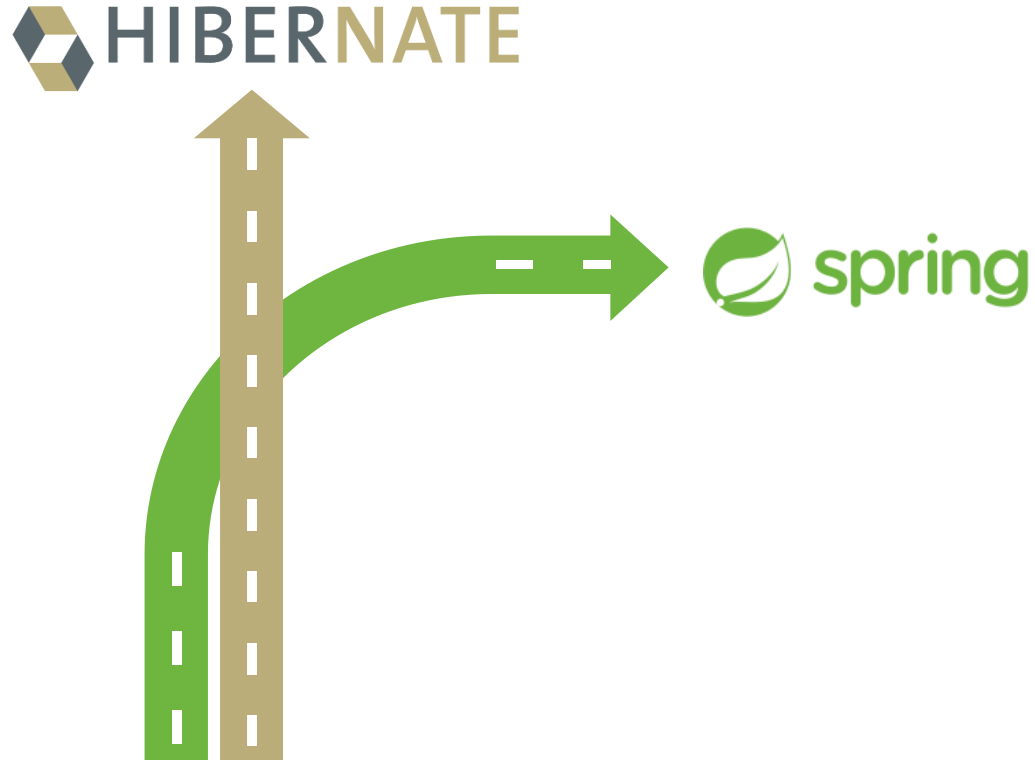
Where is Java Used?

Java has been used to develop below frameworks:



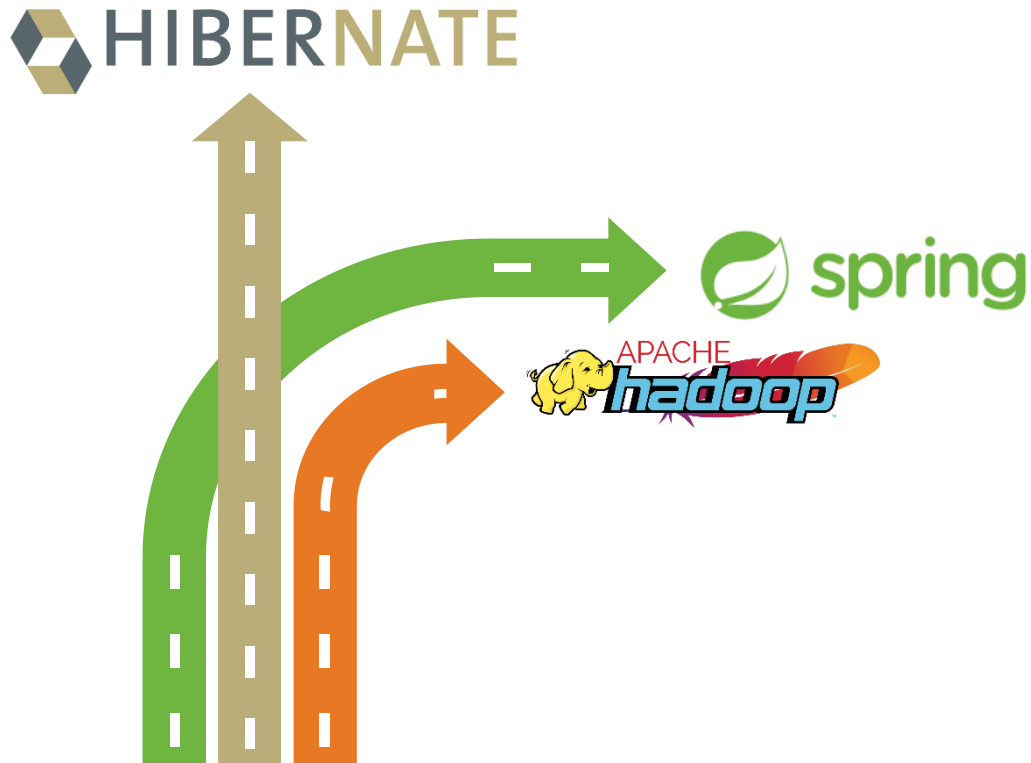
Where is Java Used?

Java has been used to develop below frameworks:



Where is Java Used?

Java has been used to develop below frameworks:

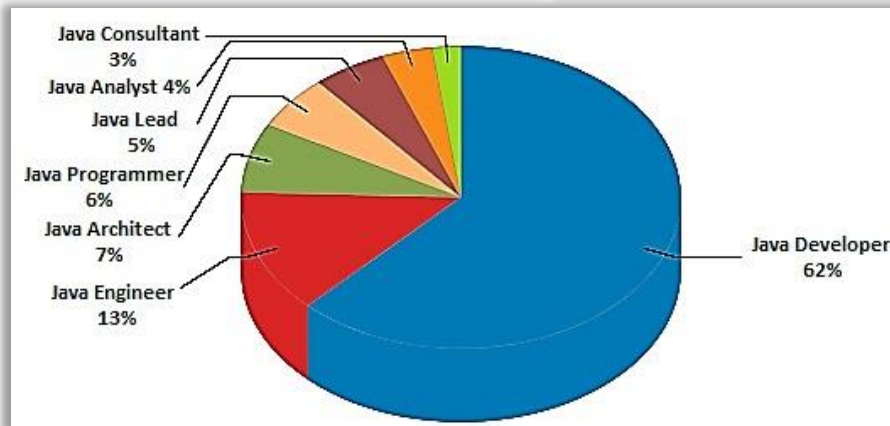
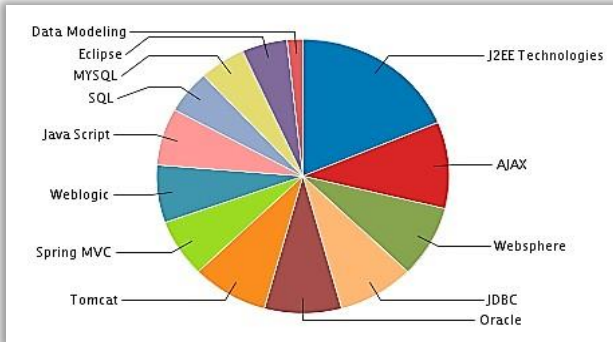
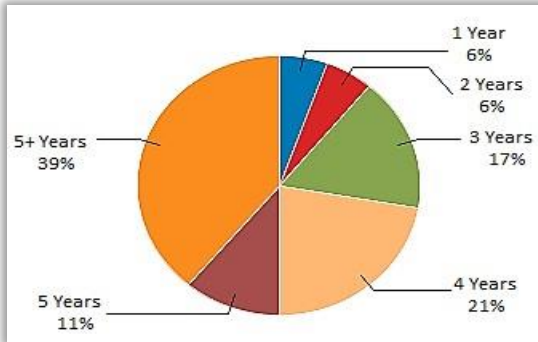
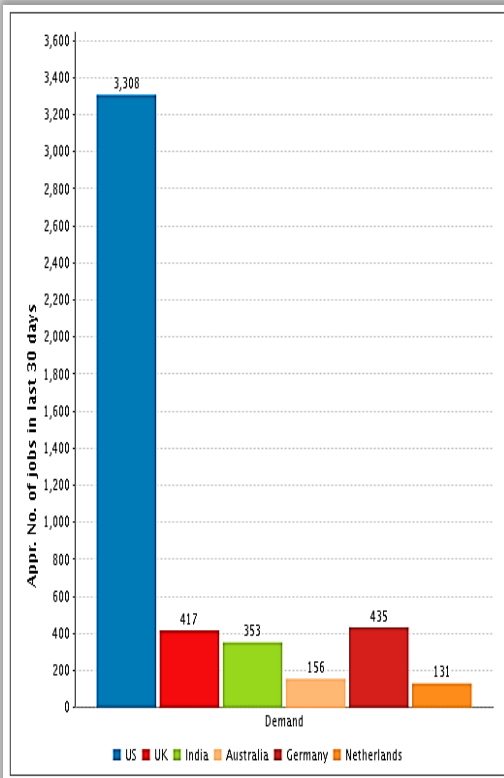


Where is Java Used?

Java has been used to develop below frameworks:



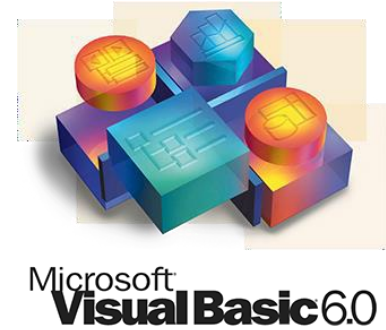
Java-Job, Demand and Experience Trends



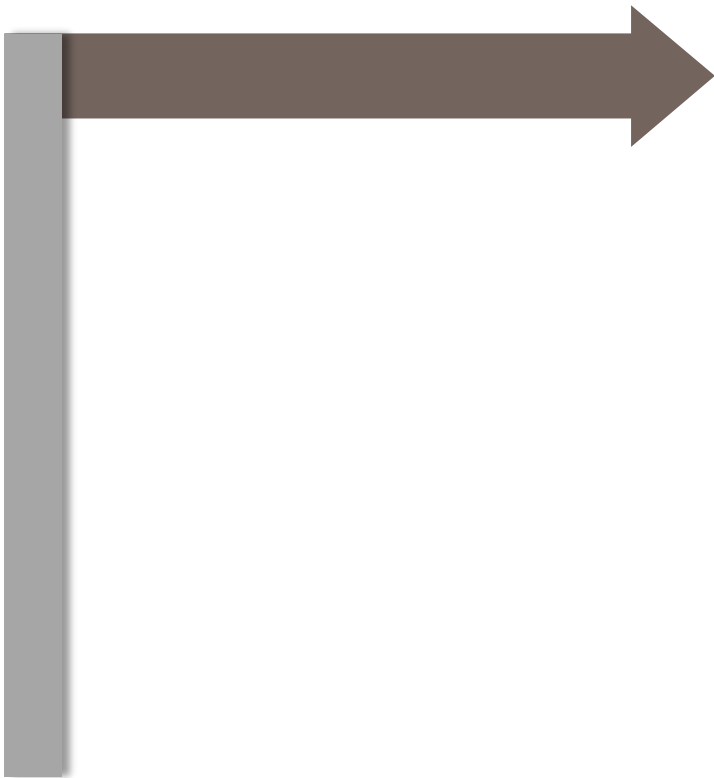
<http://jobgraphs.com/java>

Languages before Java

Before Java emerged as a programming language, there were many other programming languages:



Why not C++?



C++ does not provide efficient means for garbage collection

Why not C++?



C++ does not provide efficient means for garbage collection

No built in support for threads

Why not C++?

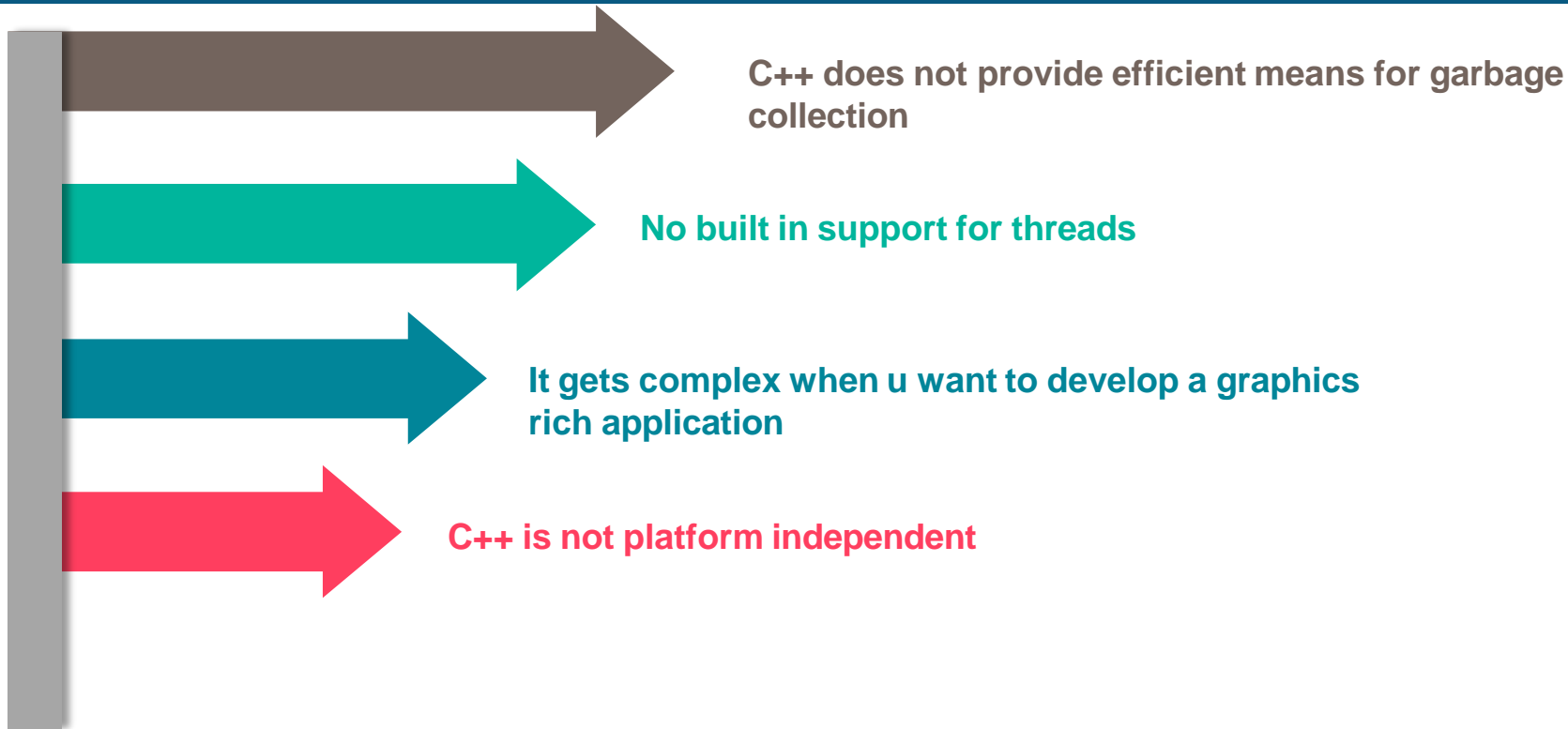


C++ does not provide efficient means for garbage collection

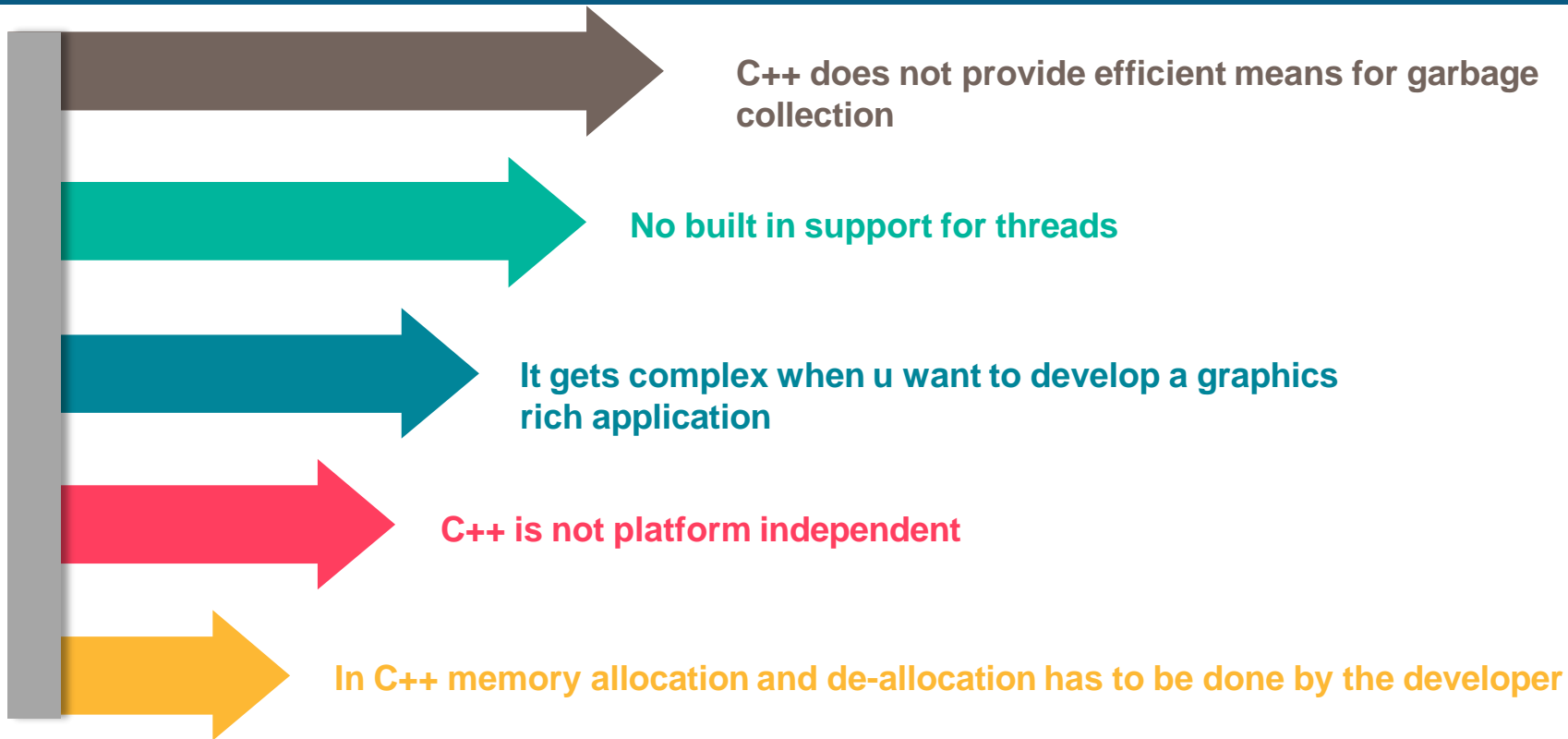
No built in support for threads

It gets complex when u want to develop a graphics rich application

Why not C++?



Why not C++?

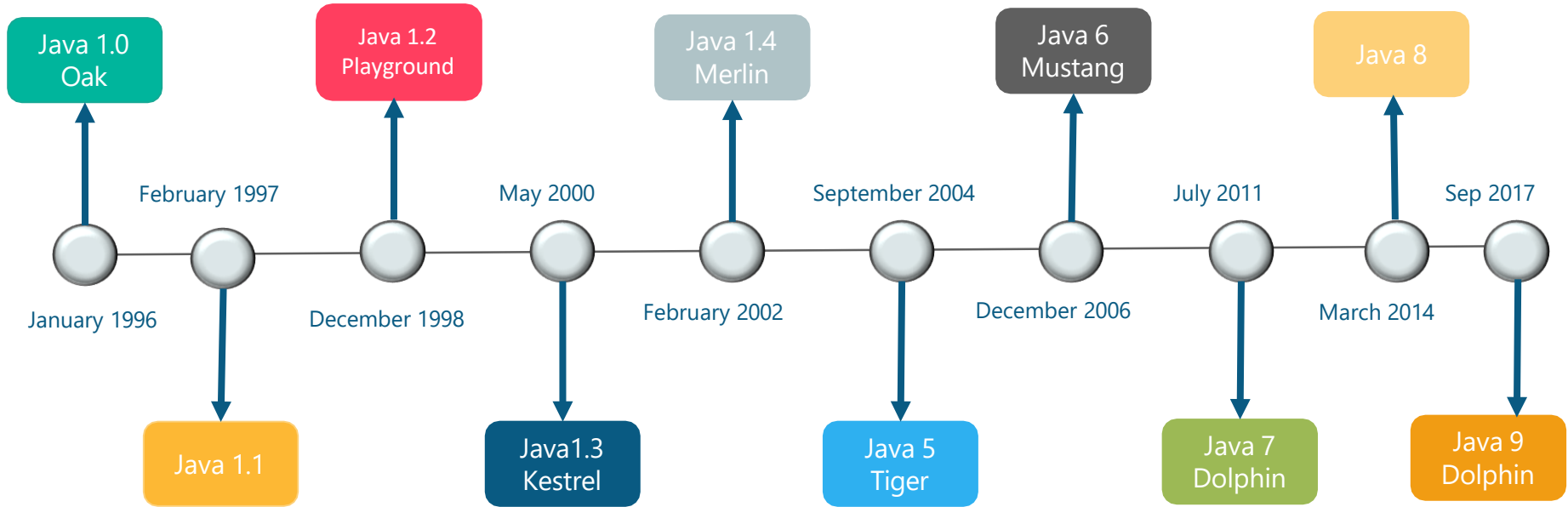


History of Java



- Java was developed by James Gosling in Sun Microsystems
- It is a platform independent programming language
- This language was initially named as OAK and later renamed as Java

Java Versions





Features of Java

Features of Java



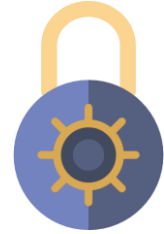
Simple



Portable



Object-oriented



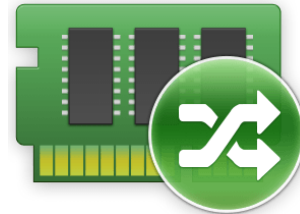
Secure



Distributed



Dynamic



Robust



HIGH PERFORMANCE

High Performance

Features of Java

- **Simple**
- Portable
- Object-oriented
- Secure
- Distributed
- Dynamic
- Robust
- High Performance

- Java was designed to be easy for professional programmer to learn and use effectively



Simple

Features of Java

- Simple
- **Portable**
- Object-oriented
- Secure
- Distributed
- Dynamic
- Robust
- High Performance

- Applications written using Java are portable in the sense that they can be executed on any kind of computer containing any CPU or any operating system



Features of Java

- Simple
- Portable
- **Object-oriented**
- Secure
- Distributed
- Dynamic
- Robust
- High Performance

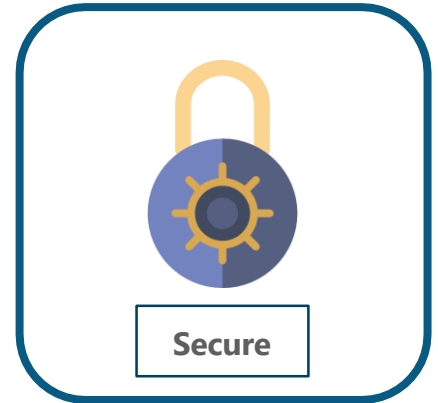
- Java is true object oriented language.
- Everything is considered to be an “object” and all operations are performed using these objects



Features of Java

- Simple
- Portable
- Object-oriented
- **Secure**
- Distributed
- Dynamic
- Robust
- High Performance

- Java Programs run inside virtual machine sandbox to prevent any activity from untrusted sources.
- No use of explicit pointer



Features of Java

- Simple
- Portable
- Object-oriented
- Secure
- Distributed**
- Dynamic
- Robust
- High Performance

- Java has a feature called Remote Method Invocation (RMI) using which a program can invoke method of another program across a network and get the output



Distributed

Features of Java

- Simple
- Portable
- Object-oriented
- Secure
- Distributed
- **Dynamic**
- Robust
- High Performance

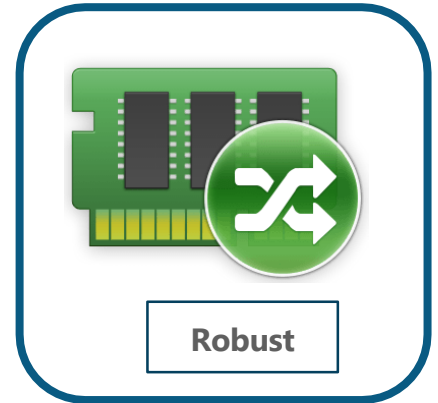
- Java programs carry with them substantial amounts of run-time type information that is used to verify and resolve accesses to objects at run time



Features of Java

- Simple
- Portable
- Object-oriented
- Secure
- Distributed
- Dynamic
- **Robust**
- High Performance

- Java checks the code during the compilation time and run time also
- Java completely takes care of memory allocation and releasing, which makes the Java program more robust



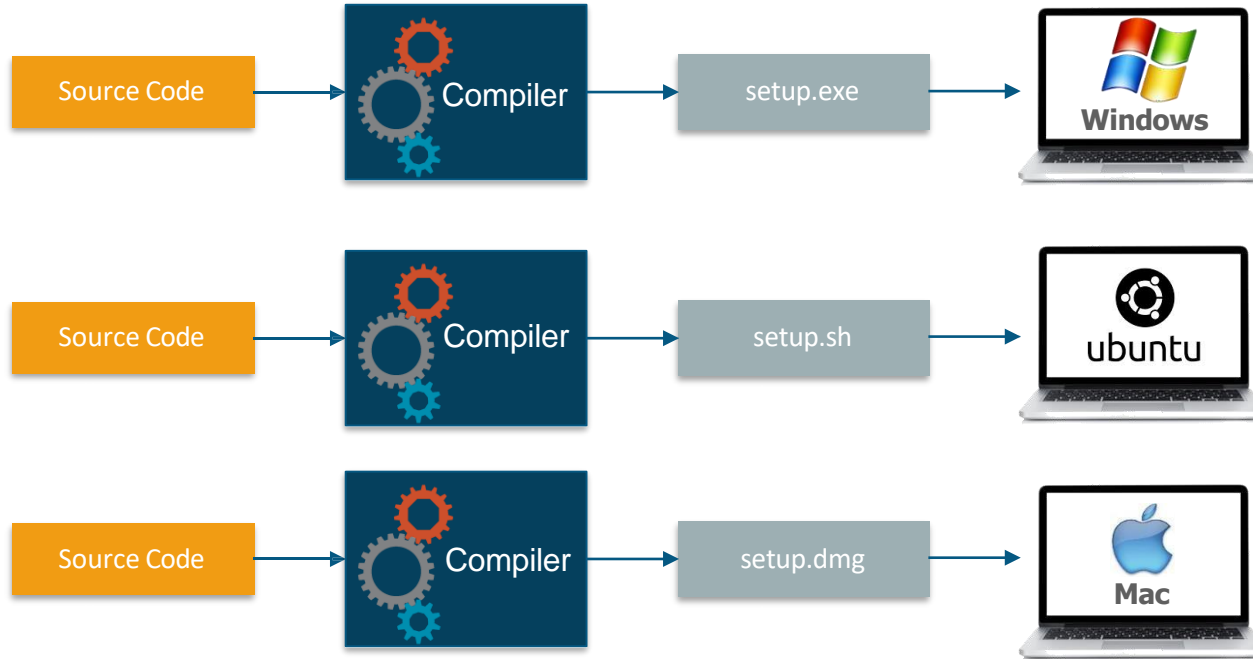
Features of Java

- Simple
- Portable
- Object-oriented
- Secure
- Distributed
- Dynamic
- Robust
- High Performance**

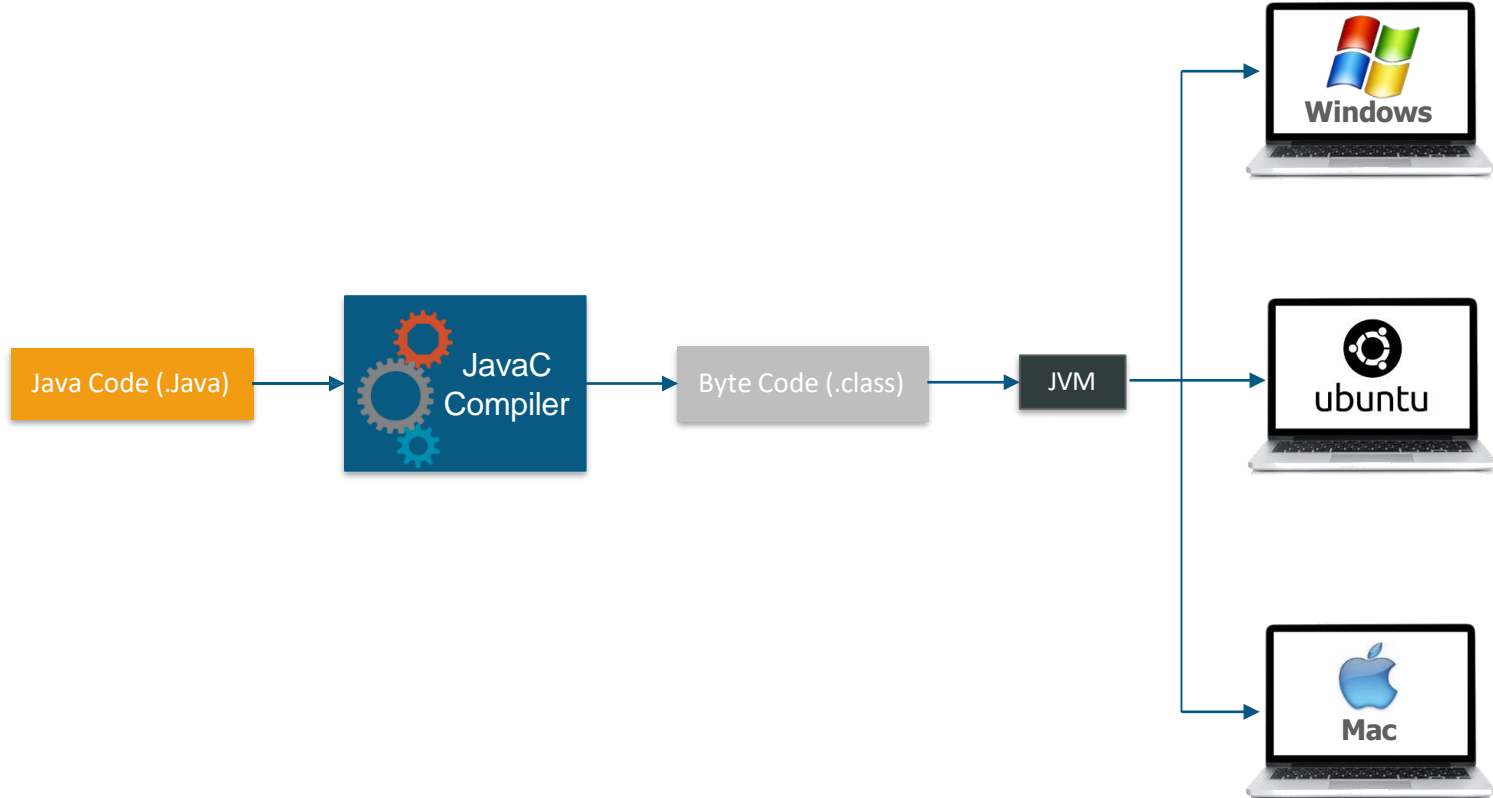
- Java achieves high performance through the use of bytecode which can be easily translated into native machine code



C++: Platform Dependent



Java: Platform Independent



In- Class Questions

1. What is bytecode?
2. What is the advantage of executing parallel threads/tasks at a time?

In Class Question - Solutions

1. What is bytecode?

Solution: Bytecode is an intermediate code which gets generated when a Java file is compiled using a Javac compiler. After compilation .class file is generated which contains the byte code. This code is platform independent.

2. What is the advantage of executing parallel threads/tasks at a time?

Solution: When many tasks/threads run at the same time, performance of the system increases.

Bytecode and Java Virtual Machine (JVM)

- Java bytecode is the form of instructions that the JVM executes
- A Java programmer need not understand bytecode at all

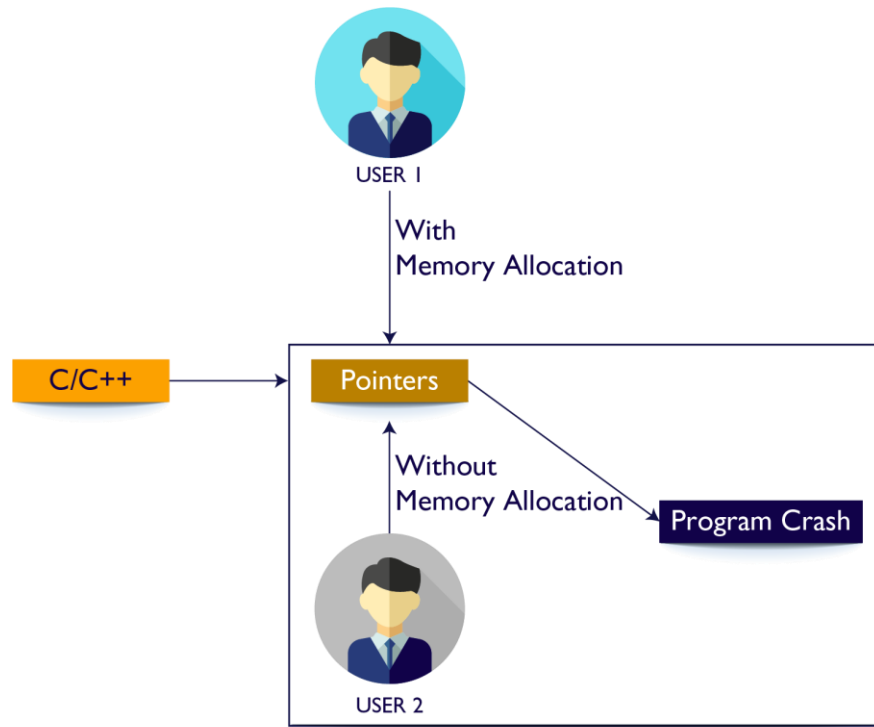
Java Virtual Machine (JVM)

- Runs the bytecode
- Makes Java platform independent
- Handles memory management



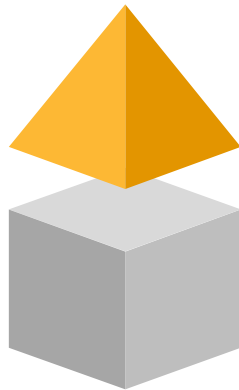
Java Memory Management

- C/C++ has pointers. User can allocate memory to these pointers
- If the pointer is accessed without allocating memory or invalid pointer is accessed then program crashes
- These issues are removed from Java as Java does not have pointers. Complete memory management is handled by Java itself



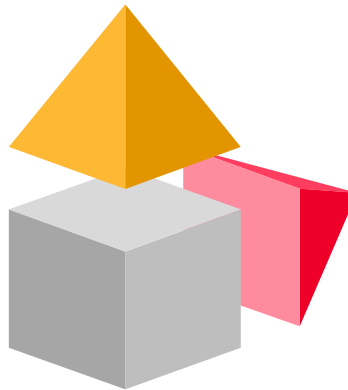
Heap and Garbage Collector

The memory area in JVM
where objects are created is
called Heap



Heap and Garbage Collector

The memory area in JVM
where objects are created is
called Heap



Heap is divided into
two parts. Young
space and old space

Heap and Garbage Collector

The memory area in JVM
where objects are created is
called Heap



Heap is divided into
two parts. Young
space and old space

The memory is freed during
runtime by a special thread
called Garbage Collector

Heap and Garbage Collector

The memory area in JVM where objects are created is called Heap



Heap is divided into two parts. Young space and old space

The memory is freed during runtime by a special thread called Garbage Collector

The Garbage Collector looks for objects which are no longer needed by the program and destroys them

Heap and Garbage Collector

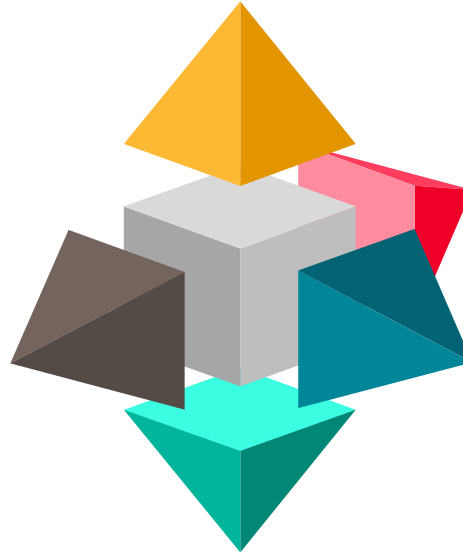
The memory area in JVM where objects are created is called Heap

Heap is divided into two parts. Young space and old space

All the newly allocated objects are created in young space. Once the young space is full then garbage collector is called so that memory can be released

The memory is freed during runtime by a special thread called Garbage Collector

The Garbage Collector looks for objects which are no longer needed by the program and destroys them



Heap and Garbage Collector

The memory area in JVM where objects are created is called Heap

If the object has lived for long in young space then they will be moved to old space. Once the old space is full, garbage collector is called to release the space in heap

All the newly allocated objects are created in young space. Once the young space is full then garbage collector is called so that memory can be released

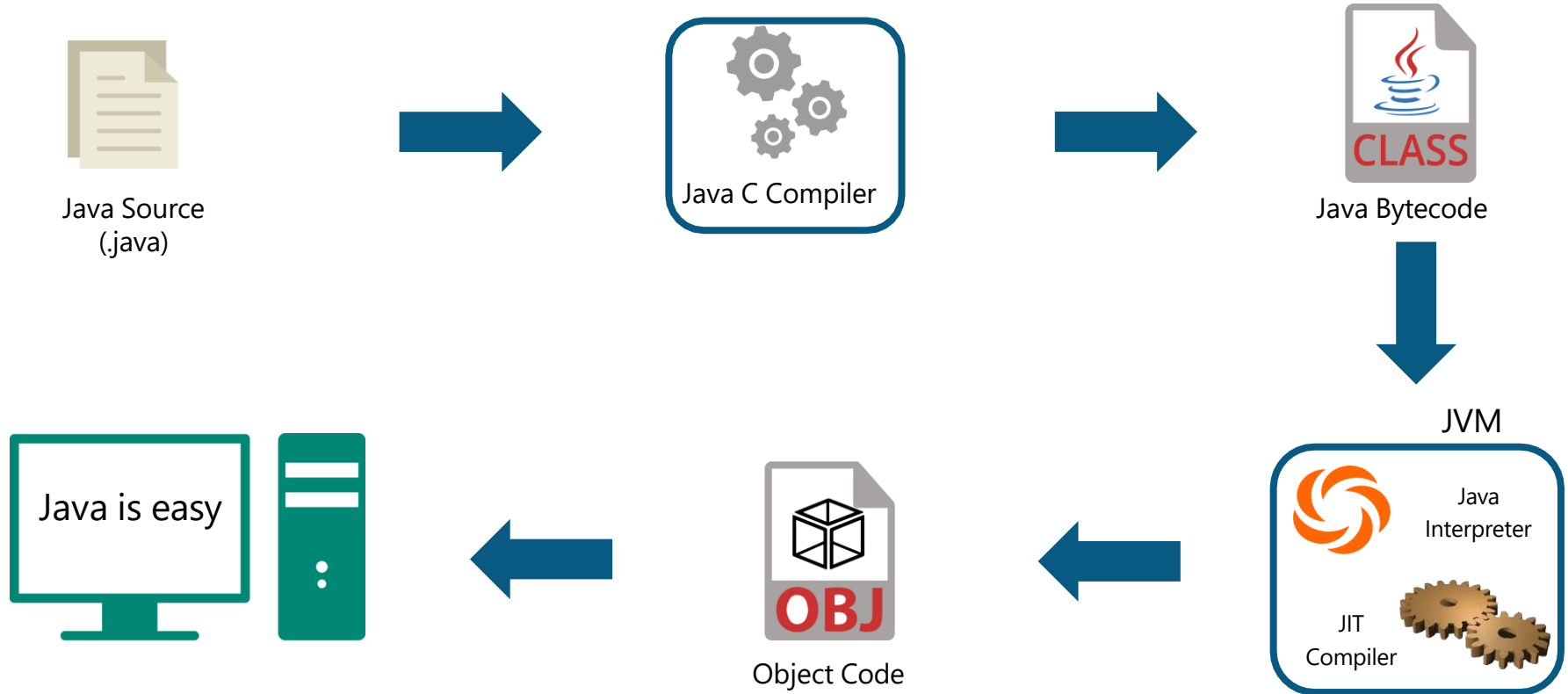


Heap is divided into two parts. Young space and old space

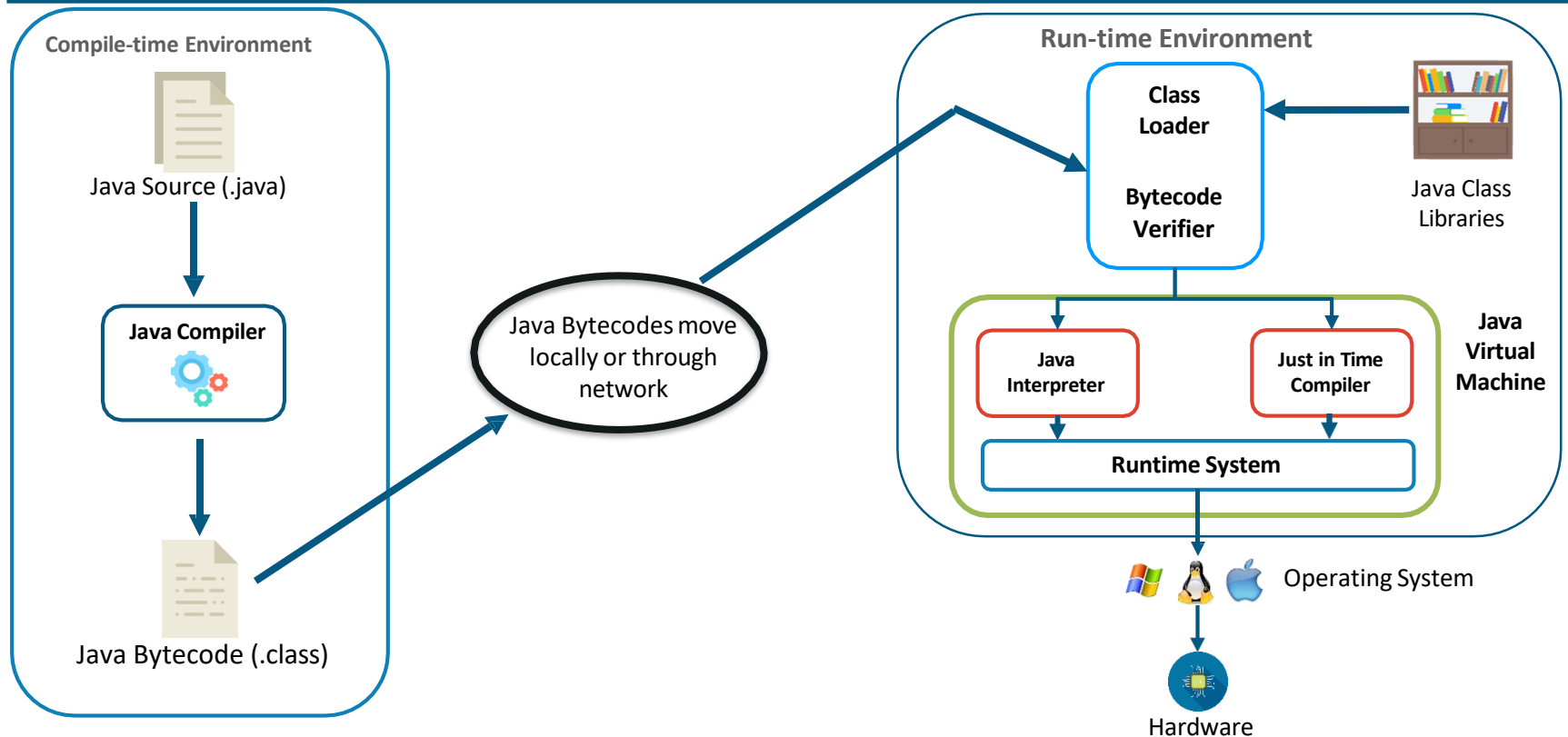
The memory is freed during runtime by a special thread called Garbage Collector

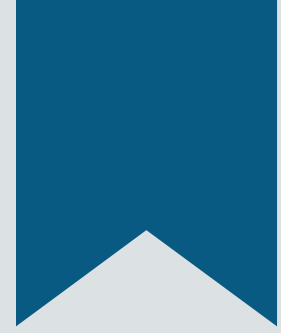
The Garbage Collector looks for objects which are no longer needed by the program and destroys them

How Java work?



How Java Works?





JAVA Installation

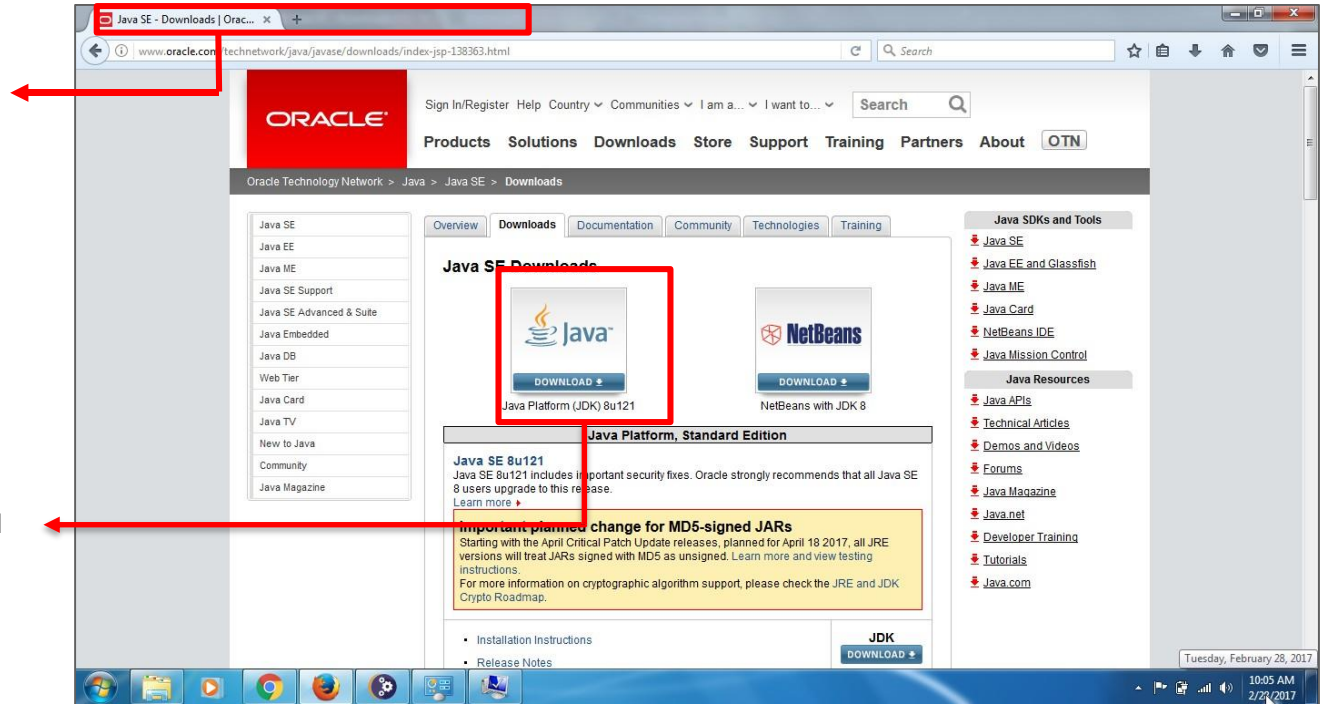
Java Installation

1

Go to
<http://www.oracle.com/technetwork/java/javase/downloads/index.html>

2

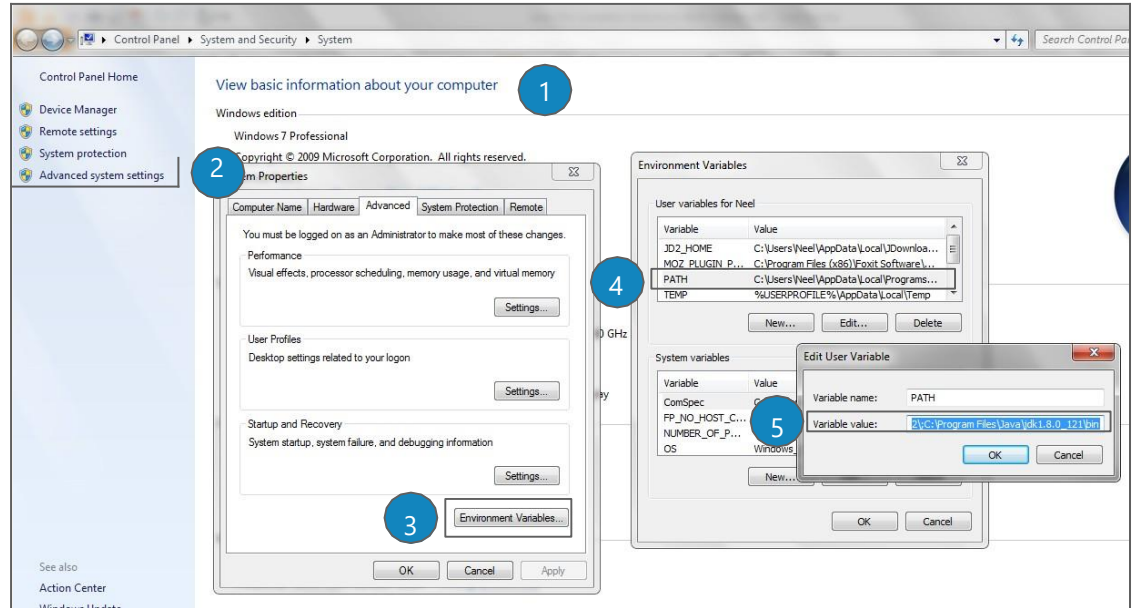
Download Java Platform(JDK) version 8 update 121 and install on your Windows system



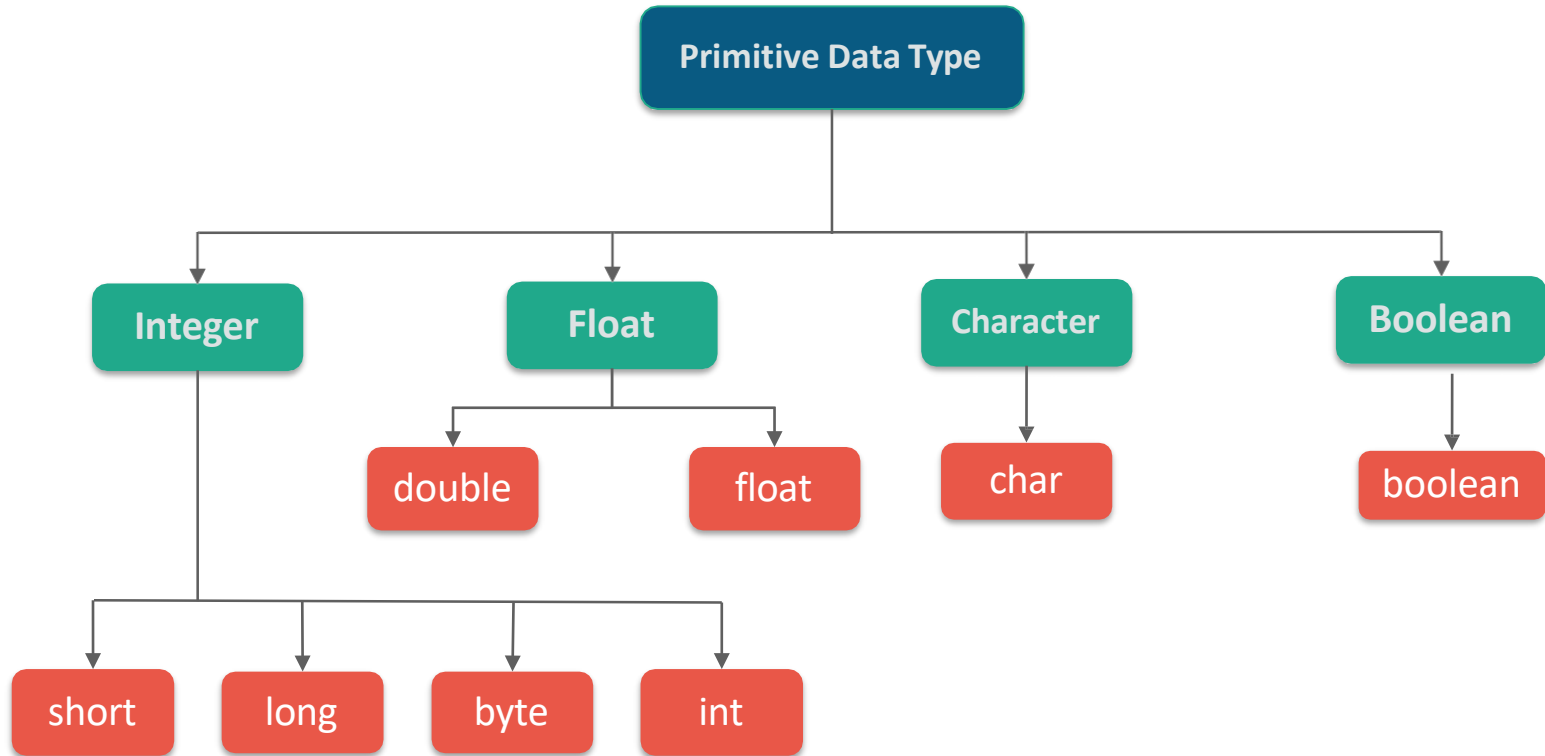
Java Installation

To set the permanent path of JDK in your system:

1. Begin by going to My Computer Properties.
2. Click on Advanced system setting.
3. Click on Environment Variables option
4. Select Path variable to edit
5. Enter the path to bin folder inside JDK installed on your system.
6. (Default path is `C:\Program Files\Java\jdk1.8.0_121\bin`)



Data Types



Data Operations

Data Operations

Arithmetic Operators

`+` → Addition
`-` → Subtraction
`*` → Multiplication
`/` → Division
`%` → Modulus

Unary Operators

`++` → Increment
operator
`--` → Decrement
operator

Relational Operators

`<` → Less than
`<=` → Less than or
equal to
`>` → Greater than
`>=` → Greater than
or equal to
`!=` → Not equal to
`==` → Double Equals

Logical Operators

`&&` → And
`||` → Or
`!` → Not



In Class Question

1. Why do you think we need byte, short and long datatypes when int can be used?

In Class Question - Solution

1. Why do you think we need byte, short and long datatypes when int can be used?

Solution: 'byte' takes 1 byte, 'short' takes 2 bytes and 'int' takes 4 bytes in memory. If we know that the variable need not be very big then 'short' is fine as it reduces the memory consumption which in turn increases the performance of the project.

Required Software

Required software for Java Programming

- JDK 1.8 64 bit or 32 bit according to your machine

<http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html>

Java SE Development Kit 8u151		
You must accept the Oracle Binary Code License Agreement for Java SE to download this software.		
<input type="radio"/> Accept License Agreement <input type="radio"/> Decline License Agreement		
Product / File Description	File Size	Download
Linux ARM 32 Hard Float ABI	77.9 MB	jdk-8u151-linux-arm32-vfp-hflt.tar.gz
Linux ARM 64 Hard Float ABI	74.85 MB	jdk-8u151-linux-arm64-vfp-hflt.tar.gz
Linux x86	168.95 MB	jdk-8u151-linux-i586.rpm
Linux x86	183.73 MB	jdk-8u151-linux-i586.tar.gz
Linux x64	166.1 MB	jdk-8u151-linux-x64.rpm
Linux x64	180.95 MB	jdk-8u151-linux-x64.tar.gz
macOS	247.06 MB	jdk-8u151-macosx-x64.dmg
Solaris SPARC 64-bit	140.06 MB	jdk-8u151-solaris-sparcv9.tar.Z
Solaris SPARC 64-bit	99.32 MB	jdk-8u151-solaris-sparcv9.tar.gz
Solaris x64	140.65 MB	jdk-8u151-solaris-x64.tar.Z
Solaris x64	97 MB	jdk-8u151-solaris-x64.tar.gz
Windows x86	198.04 MB	jdk-8u151-windows-i586.exe
Windows x64	205.95 MB	jdk-8u151-windows-x64.exe

Required Software (Contd.)

Required software for Java Programming

- Eclipse J2EE version for 64 bit or 32 bit according to your machine

<https://www.eclipse.org/downloads/>

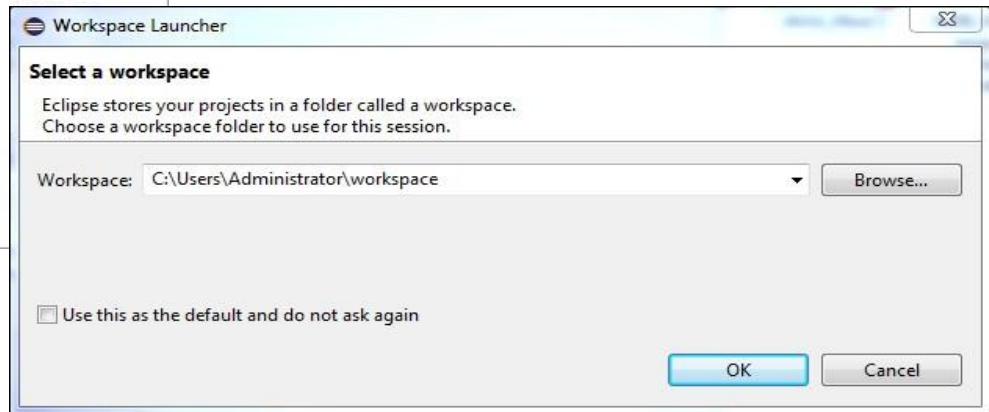
A rectangular orange button with rounded corners. It contains a white download icon (a square with a downward arrow) followed by the word "DOWNLOAD" in white, uppercase, sans-serif font.

Download from: Japan - Japan Advanced Institute of Science and Technology ([http](http://www.eclipse.org/downloads/))

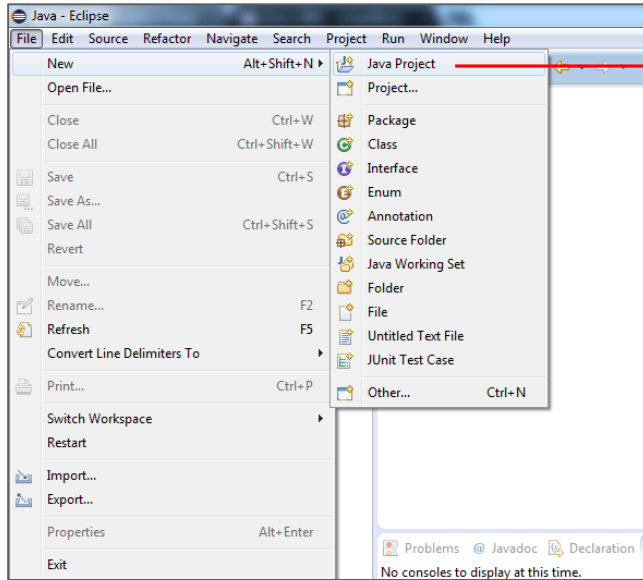
File: `eclipse-inst-win64.exe` SHA-512

>> Select Another Mirror

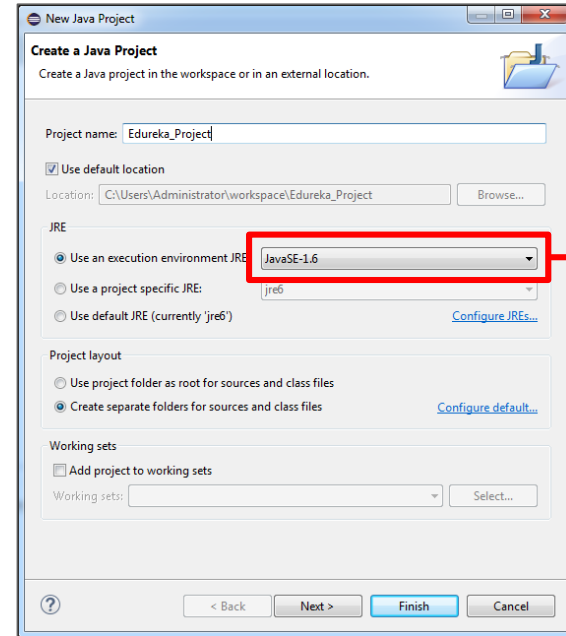
Select a Workspace



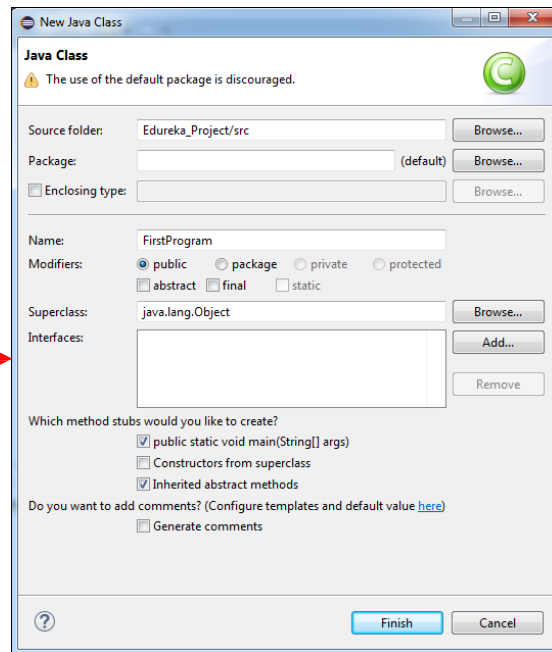
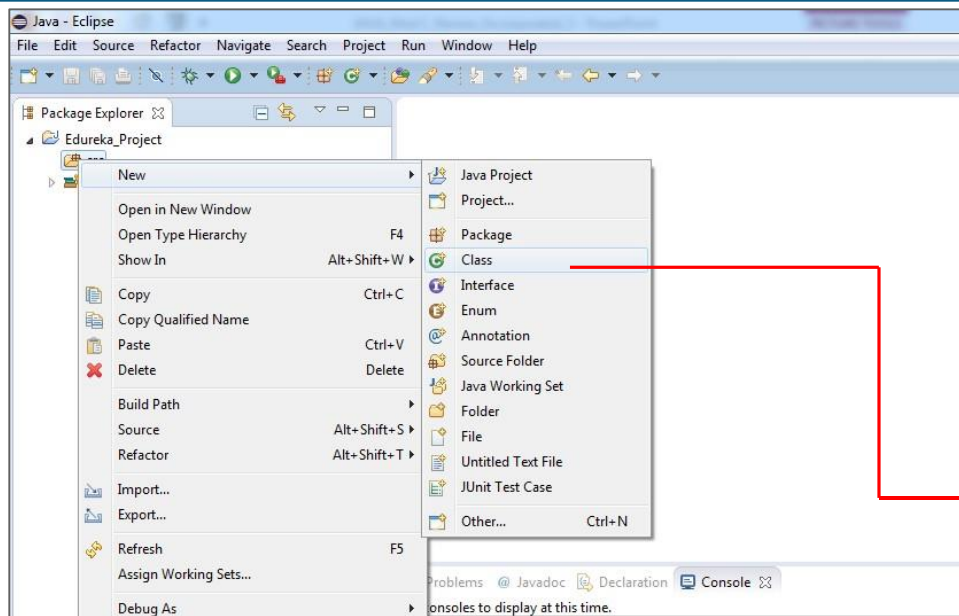
Create a Project



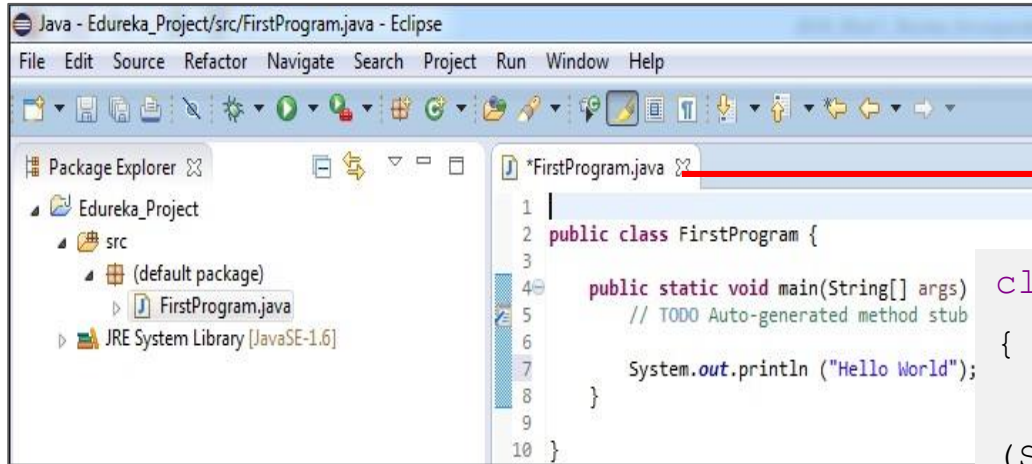
Select the version you want to work in



Create a Class



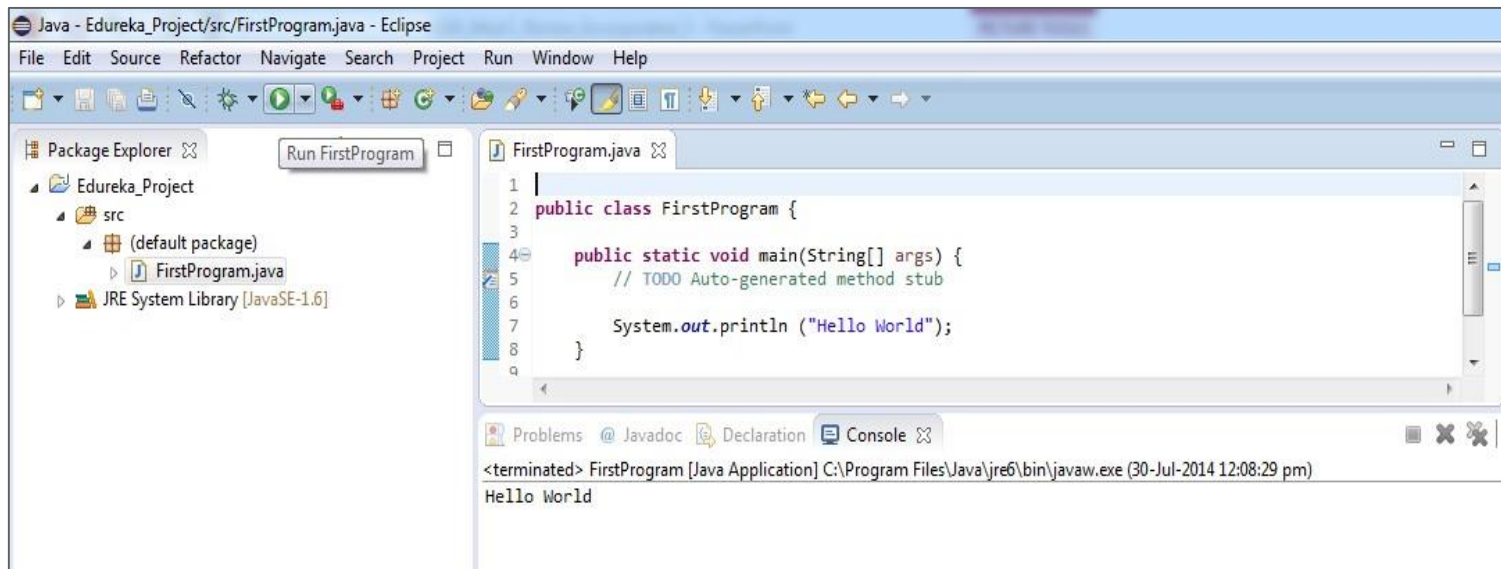
My First Java Program



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

Execute java Program

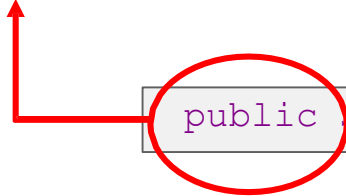
- Short-cut → ctrl + F11



Main Method

- Every program of Java should be inside a class with a class name

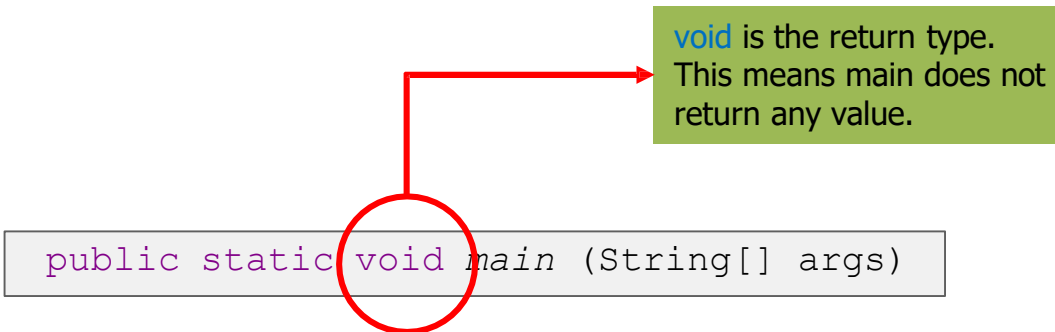
`public` means this function is accessible globally across all the programs.



```
public static void main (String[] args)
```

Main Method

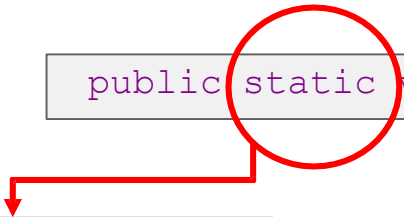
- Every program of Java should be inside a class with a class name



Main Method

- Every program of Java should be inside a class with a class name

```
public static void main (String[] args)
```



static means without creating the object of a class, you can access this method. You will study more about static in 3rd module of Java

Main Method

- Every program of Java should be inside a class with a class name

```
public static void main (String[] args)
```



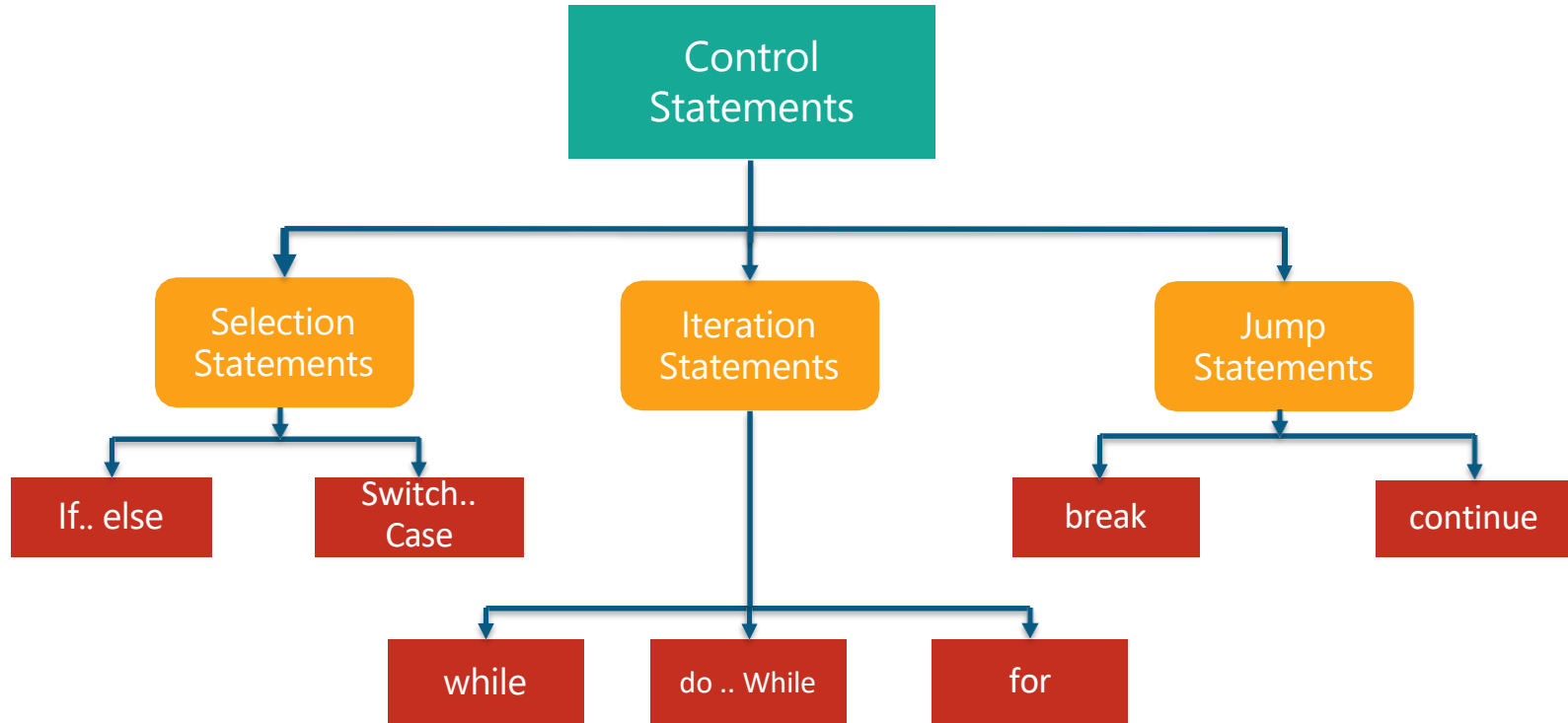
main is the function which will be called by JVM and String[] args is the command line arguments.

Programs for Data Types and Operations

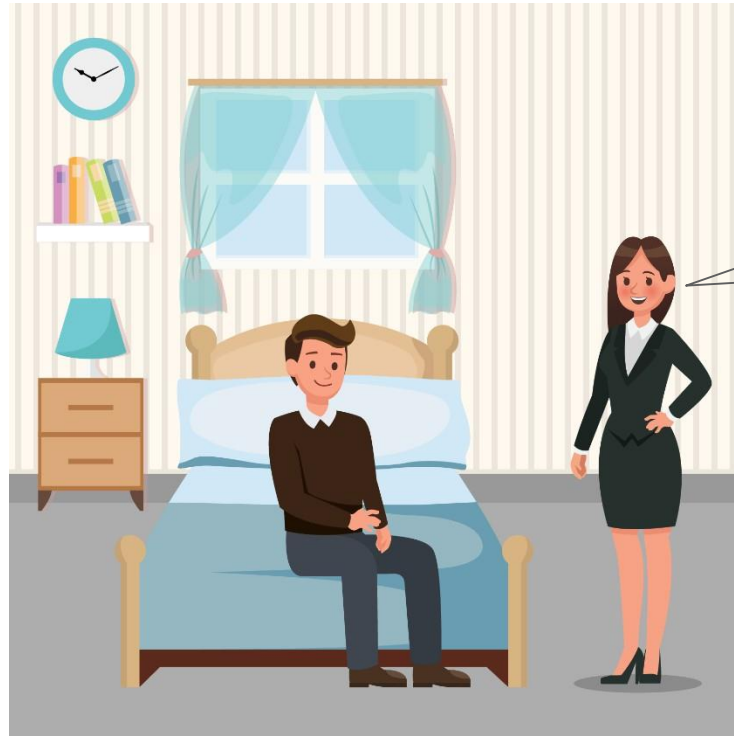
- Program to add two numbers

```
class FirstProgram
{
    public static void main (String[] args)
    {
        int a = 20, b = 30, c;
        c = a + b;
        System.out.println ("Result is : " + c);
    }
}
```

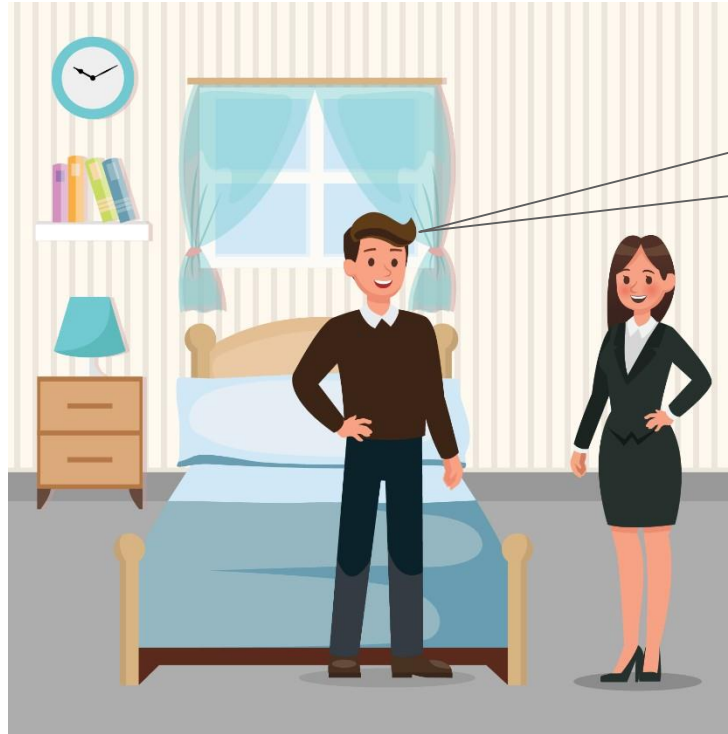
Choices in Real life



Choices in Real life



If-else Condition



Only, If I come back
from office at 5 or else
we shall go tomorrow.

Control Statement

- Selection Statements

- Iteration Statements

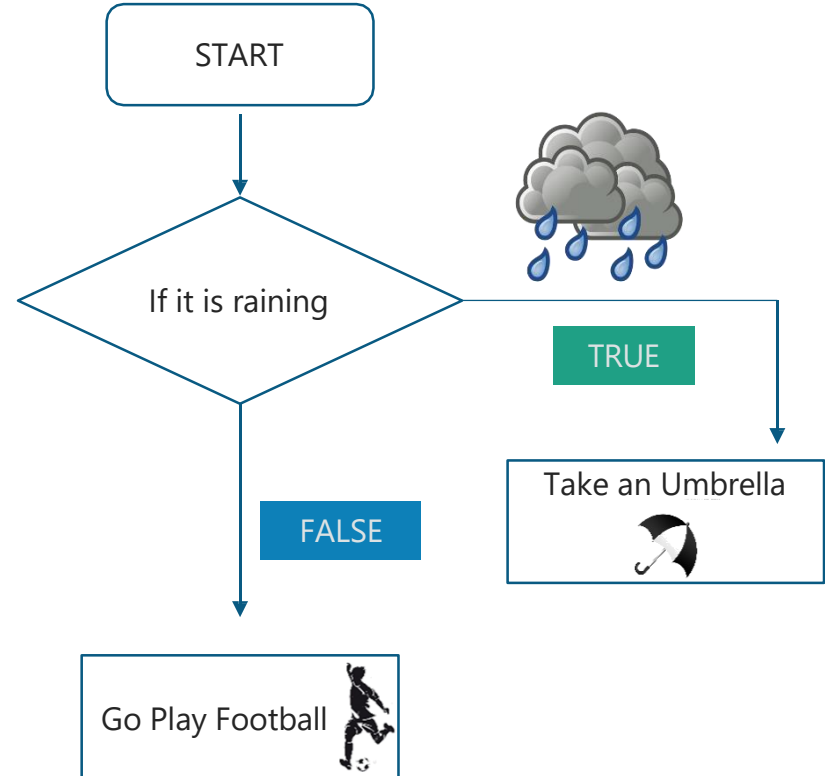
- Jump Statements

1. If statement

Syntax:

```
if (condition):  
    statements 1 ...
```

```
else:  
    statements 2 ...
```



Program on if Condition

```
class testif
{
    public static void main (String[] args)
    {
        int balance = 10000;
        boolean withdraw;

        if (balance < 0)
            withdraw = false;
        else
            withdraw = true;

        System.out.println ("Can I withdraw : " + withdraw);
    }
}
```


Control Statement

Selection
Statements

Iteration
Statements

Jump
Statements

2. Switch statement

Syntax:

```
switch (expression) {
```

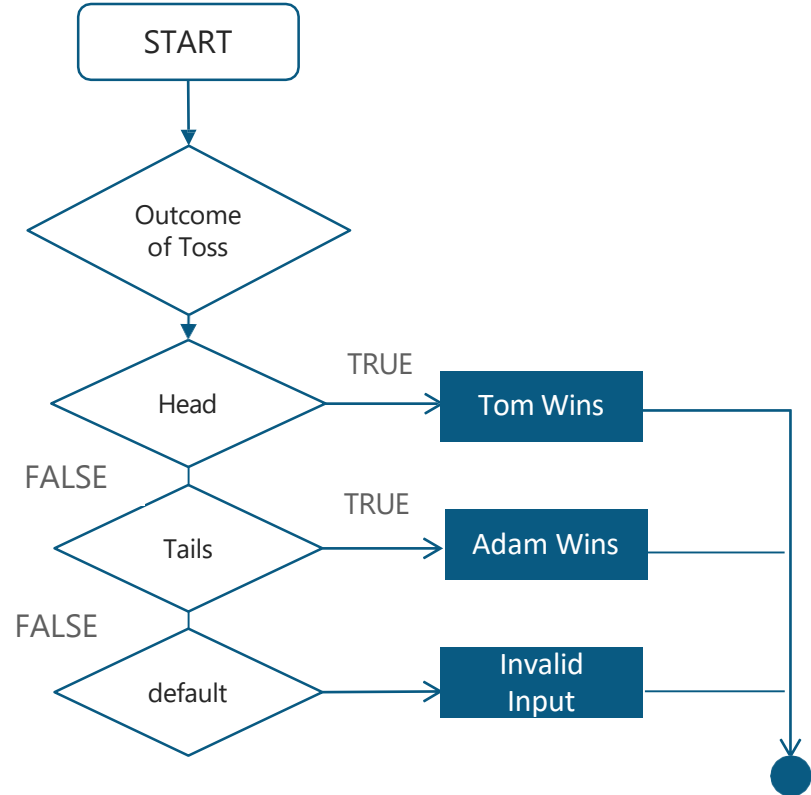
```
case value1: Statement1
```

```
break;
```

```
case value2: Statement2
```

```
break;
```

```
.  
.
```



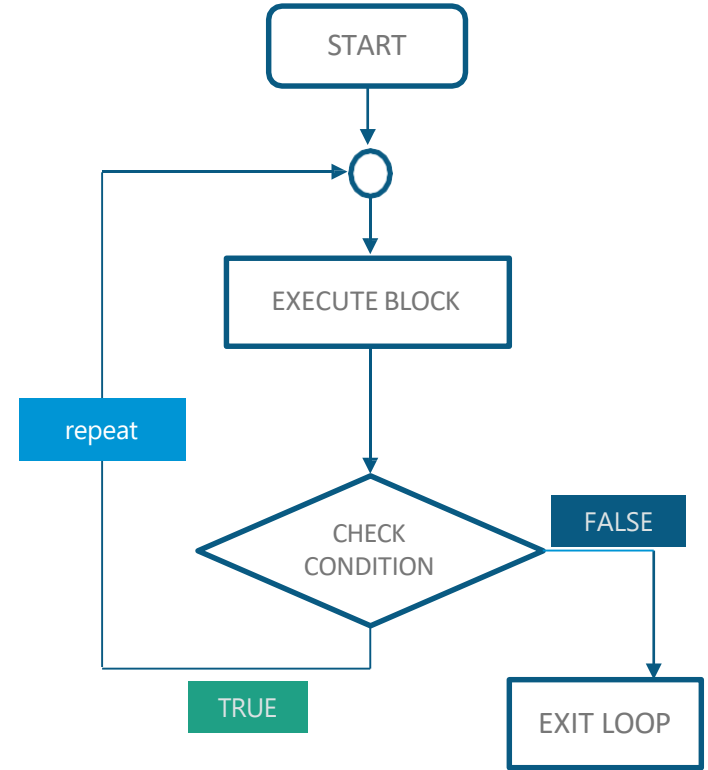
Control Statement

- Selection Statements
- Iteration Statements
- Jump Statements

3. do while statement

Syntax:

```
do
{
statements...
} while (condition is True)
```



Control Statement

Selection
Statements

Iteration
Statements

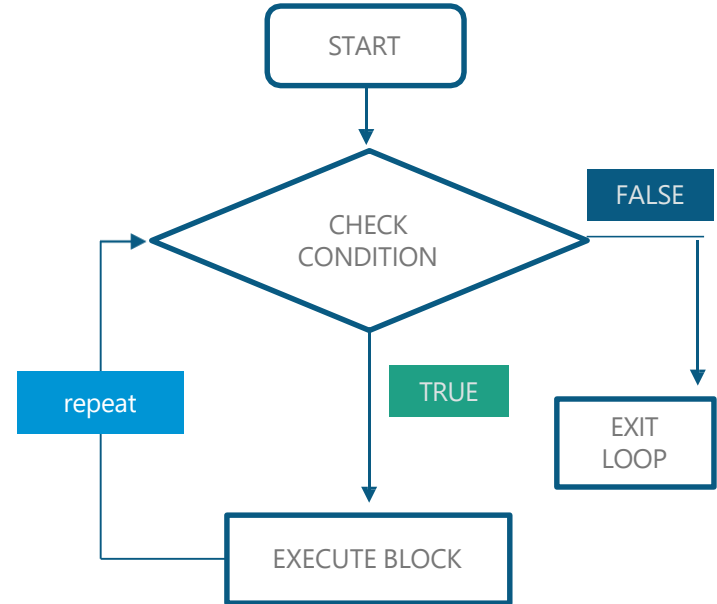
Jump
Statements

4. while statement

Syntax:

`while (condition is True)`

```
{  
statements...  
}
```



while-Loop : Program

- Program to print numbers from 1 to 10

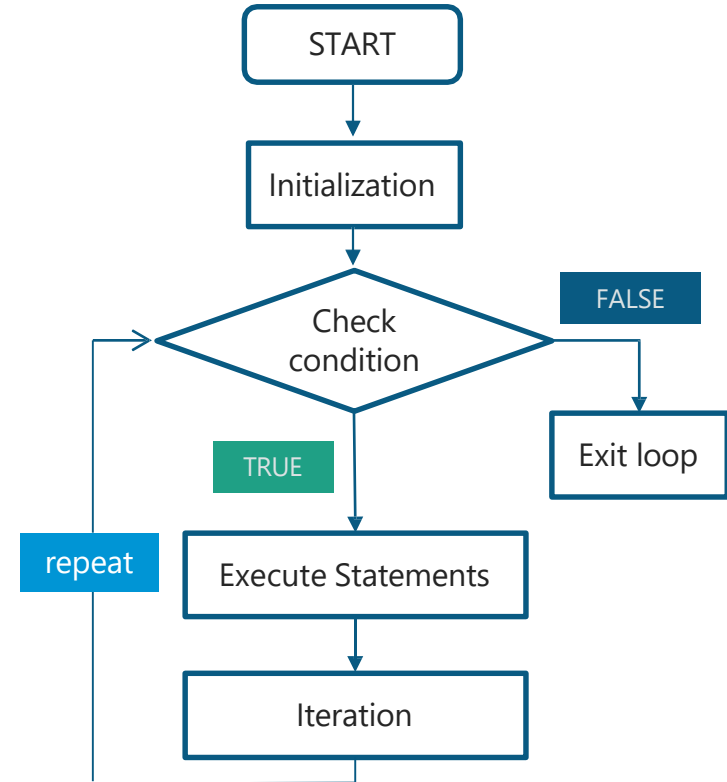
```
class WhileLoopDemo {  
    public static void main(String[] args) {  
  
        int i = 1;  
  
        while (i <= 10) {  
            System.out.println(i);  
            i++;  
        }  
    }  
}
```

Control Statement

- Selection Statements
- Iteration Statements
- Jump Statements

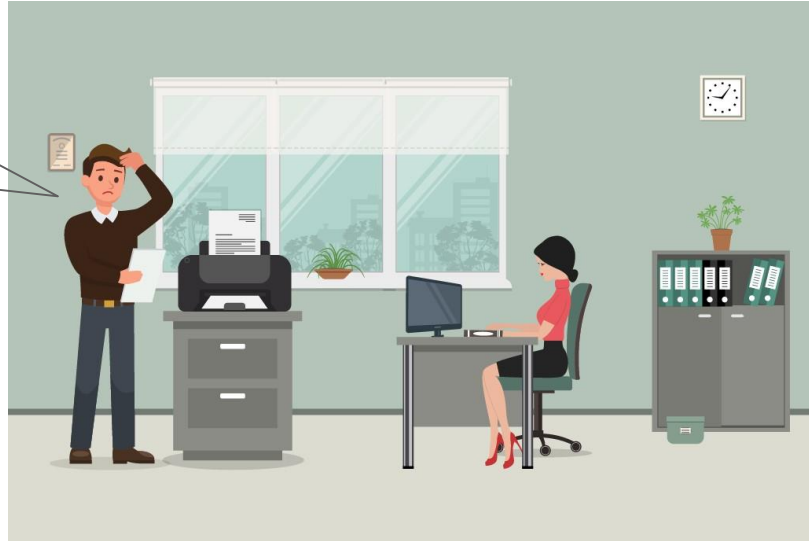
5. for statement

Syntax:
`for(initialization; condition;
iteration)
{
statements...
}`

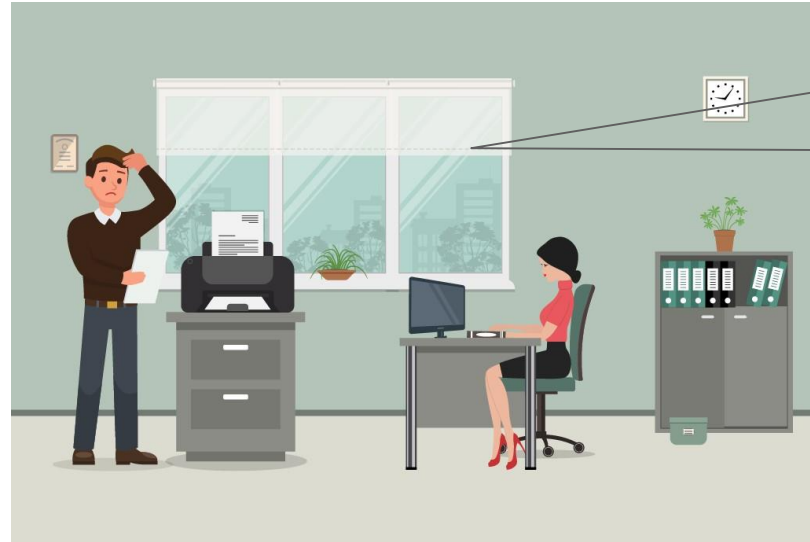


Repetitive Tasks in Real life

This is annoying. I need to take 100 printouts in 10 mins and I have to wait to press the button everytime it prints.



Repetitive Tasks in Real life(contd.)



This is annoying. I need to take 100 printouts in 10 mins and I have to wait to press the button everytime it prints.

for-Loop : Program

- Program to print numbers from 1 to 10

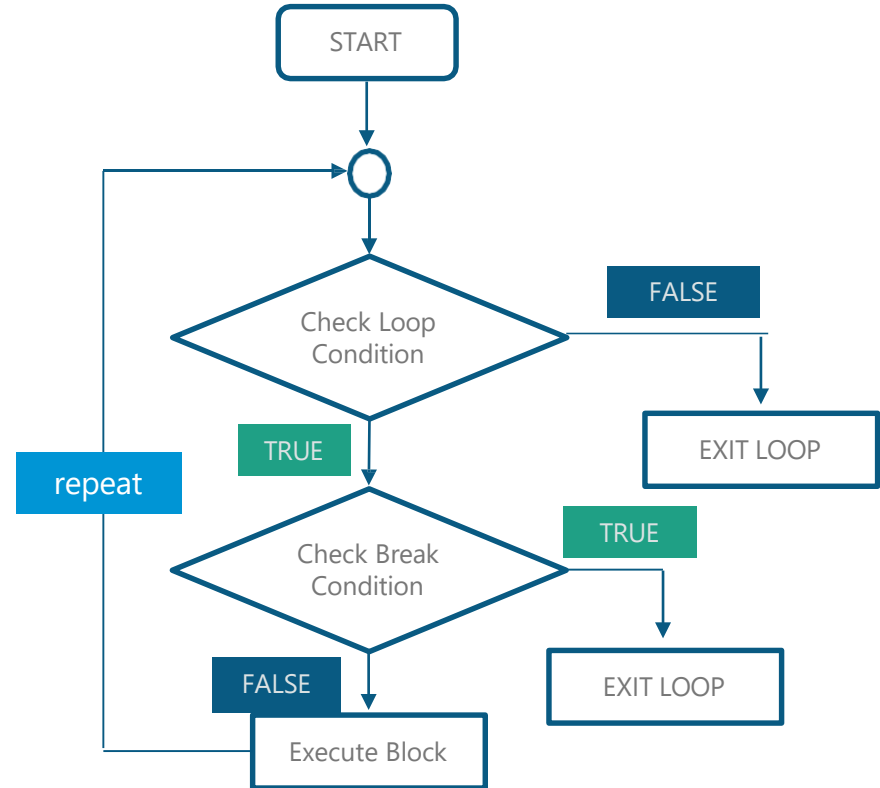
```
class ForLoopDemo
{
    public static void main (String[] args)
    {
        for (int i = 1; i <= 10; i++)
            System.out.println(i);
    }
}
```


Control Statement

- Selection Statements
- Iteration Statements
- Jump Statements

6. break

Syntax:
`break;`



Control Statement

● Selection Statements

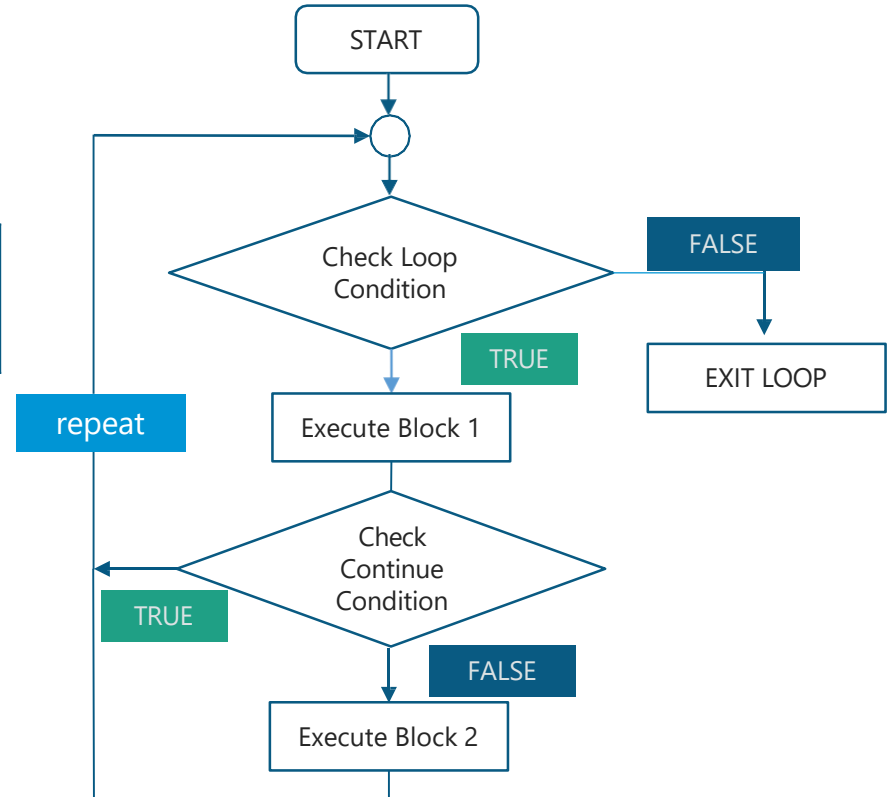
● Iteration Statements

● Jump Statements

7. continue

Syntax:

`continue;`



Assignment – Data Types and Operations

- Write programs to use all the data types and given arithmetic operations
- Write program to perform all the arithmetic operations given in the table

Arithmetic Operators	
+	→ Addition
-	→ Subtraction
*	→ Multiplication
/	→ Division
++	→ Increment operator
--	→ Decrement operator

Assignment – if Condition

if condition

- Write a program to check if the candidate is eligible for voting or not. (Hint: Check age)
- Write a program to check if the number is positive or negative
- Extend the previous program to check whether the given number is positive, zero or negative

(Hint: use if-else conditions)

- Write a program to find largest of two numbers
- Write a program to check given number is even or odd

(Hint: use % operator)

Assignment – for-loop

- Write a program to print 10 even numbers and 10 odd numbers
- Write a program to find factorial of a number.
- Write a program to generate tables of 10
- Write a program to add the digits of a number
- Write a program to reverse the digits of a number
- Write a program to generate 10 Fibonacci numbers

Assignment – while-loop

- Write a program to print 10 even numbers and 10 odd numbers
- Write a program to find factorial of a number
- Write a program to generate tables of 10
- Write a program to add the digits of a number
- Write a program to reverse the digits of a number
- Write a program to generate 10 Fibonacci numbers

Assignment – do-while loop

- Write a program to print 10 even numbers and 10 odd numbers
- Write a program to find factorial of a number
- Write a program to generate tables of 10
- Write a program to add the digits of a number
- Write a program to reverse the digits of a number
- Write a program to generate 10 Fibonacci numbers

Thank You!