GDSE-61

- **Background into Programming**
- **Java Literals**
- **Java Data types**

T.Rashmi Sharmila

What is a programming language?

A programming language is a vocabulary and set of grammatical rules for instructing a computer or computing device to perform specific tasks.

What is a java programming language?

Java is an object oriented high level programming language.

JDK-(JAVA DEVELOPMENT KIT)

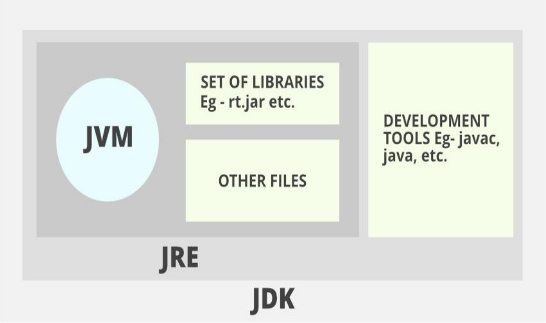
A software development environment that provides a collection of tools and libraries needed to develop a Java application.

JRE-(JAVA RUNTIME ENVIROMENT)

A set of components for creating and running a Java application.

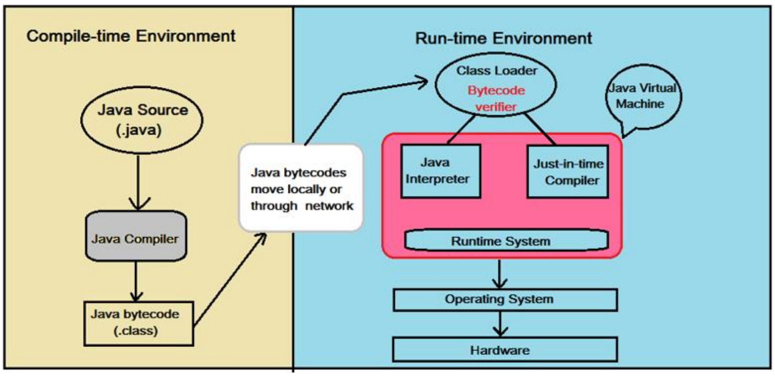
JVM-Java Virtual Machine

JVM generates a .class(Bytecode) file, and that file can be run in any OS, but JVM should have in OS because JVM is platform dependent.

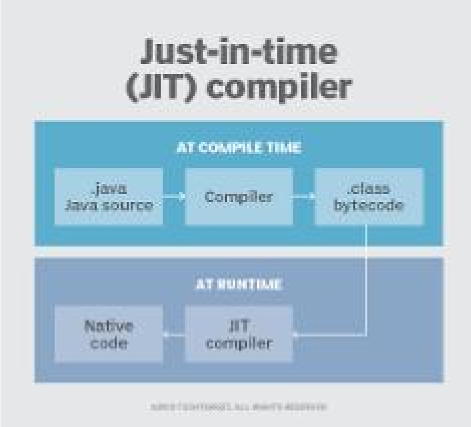

JVM
SET OF LIBRARIES Eg - rt.jar etc.
OTHER FILES
DEVELOPMENT TOOLS Eg- javac, java, etc.
JRE
JDK

Compiler and Interpreter.

The source code is completely converted to byte code by the compiler at once. The byte code file is converted from JVM to line by line machine code by interpreter.



JIT-Just-In-Time



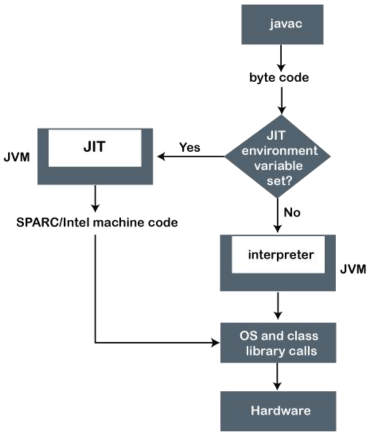
The Just-In-Time (JIT) compiler is a component of the Java Runtime Environment that improves the performance of Java applications at run time.

Why does JVM use JIT?

Java source code is compiled into class files, which contain bytecode. Since the execution of bytecode is slower than the execution of machine language code because JVM first needs to translate bytecode into machine language code. JIT helps JVM here by compiling currently executing byte code into machine language**.**

What is the difference between JIT and interpreter?

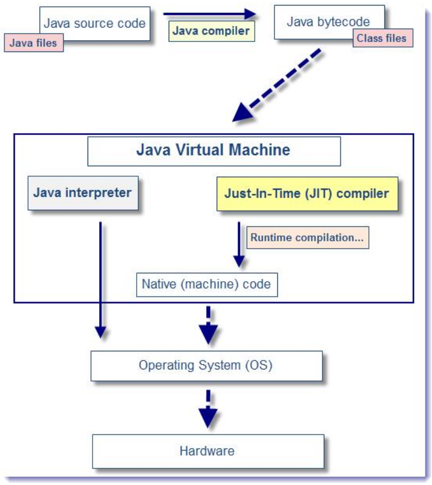
The main difference between Interpreter and JIT compiler is that the interpreter is a software that converts the source code into native machine code line by line while JIT compiler is a component in JVM that improves the performance of Java programs by compiling bytecodes into native machine codes at runtime.



JIT Compilation Process

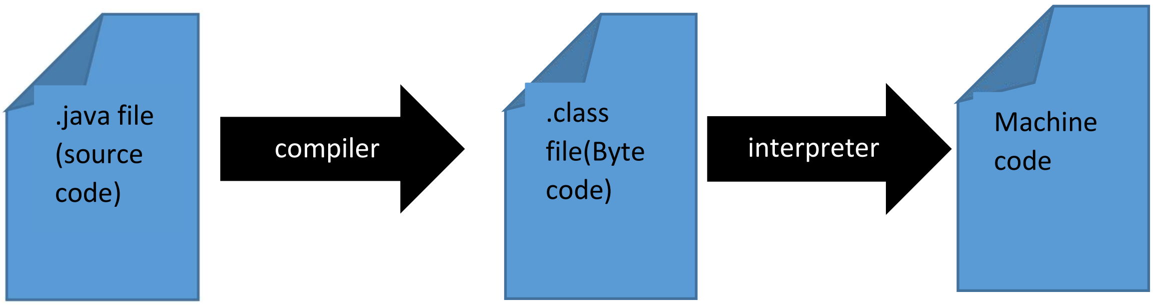
Why is JIT faster than interpreter?

A JIT compiler only looks at the bytecode once1, and compiles it to native code which can then be understood directly by the computer - no further translation required. The translation takes time, so if you can do it just the once, it's more efficient.



Machine code VS Byte code

Machine code	Byte code
The computer can understand the machine code directly.	The created code generated after compiling the source code is the byte code.



Native code

Native code compiler for Java translates the Java code into a binary representation that can be linked to precompiled library files and resources to create an executable program. Native code compilers eliminate the need for JVM and interpreters to convert the Java byte code, which is a portable intermediate code.

Valid main method declarations

```
1.class Example{
     public static void main(String args[]){
           System.out.println("Hallo java");
           }
     }

2. class Example{
     static public  void main(String args[]){
           System.out.println("Hallo java");
           }
     }
3. class Example{
     static public  void main(String[] args){
           System.out.println("Hallo java");
           }
     }
4. class Example{
     static public  void main(String[] rashmi){
           System.out.println("Hallo java");
           }
     }
5. class Example{
     public static   void main(String[] rashmi){
           System.out.println("Hallo java");
           }
     }
6. class Example{
     public static   void main(String rashmi[] ){
           System.out.println("Hallo java");
           }
     }
```

Print hello java

Invalid (compile ok , runtime error

```
7. class Example{
      public static  void main(String args[] ){
            System.out.println("Hallo java");
            }
      }
class Example{
      static  void main(String args[] ){
            System.out.println("Hallo java");
            }
      }
```

```
8. class Example{
      public static void main(String args[] ){
            System.out.println("Hallo java");
            }
      }
class Example{
      public  void main(String args [] ){
            System.out.println("Hallo java");
            }
      }
```

```
9. class Example{
      public static void main(String args []  ){
            System.out.println("Hallo java");
            }
      }
class Example{
      public static void main(String args   ){
            System.out.println("Hallo java");
            }     }
```

```
10.
class Example{
          public static void main(String args[]){
                System.out.println("Hallo java");
                }
}
class Example{
          public static void main(){
                System.out.println("Hallo java");
                }
}
```

```
11.
class Example{
       public static void main(String  args[] ){
              System.out.println("Hallo java");
               }
}
class Example{
       public static void Main(String  args[] ){
              System.out.println("Hallo java");
               }
}
```

Missing word

Error: Main method not found in class Example, please define the main method as:

public static void main(String[] args)

or a JavaFX application class must extend javafx.application.Application

Illegal main method declarations

```
12. class Example{
            public static void main(String  args[] ){
                     System.out.println("Hallo java");
                     }
}
class Example{
            public static main(String  args[] ){
                     System.out.println("Hallo java");
                     }
}
```

Example.java:2: error: invalid method declaration; return type required.

```
13.
class Example{
     public static void main(String args [] ){
                 System.out.println("Hallo java");
                 }
}
class Example{
     public static void main(String  [] ){
                 System.out.println("Hallo java");
                 }
}
```

Example.java:2: error: <identifier> expected

```
14. class Example{
     public  static  void  main(String  args[] ){
                 System.out.println("Hallo java");
                 }
}
class Example{
     public static void main(String  args ){
                 System.out.println("Hallo java");
                 }
}
```

Example.java:2: error: invalid method declaration; return type required

Example.java:2: error: '(' expected

Example.java:2: error: <identifier> expected

```
15.
class Example{
     public  static  void  main(String  args[] ){
                 System.out.println("Hallo java");
                 }
}
class Example{
     public  static void    main(String  args ){
```

Example.java:2: error: invalid method declaration; return type required

Example.java:2: error: '(' expected

Example.java:2: error: <identifier> expected

```
                 System.out.println("Hallo java");
                 }
}
```

16.

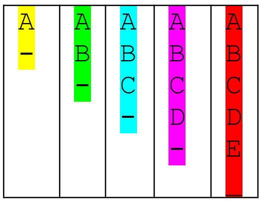
```
class Example{
     public  static  void  main(String  args[] ){
                 System.out.println("Hallo java");
                 }
}
```

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

```
Example.java:2: error: invalid method
declaration; return type required

Example.java:2: error: '(' expected

Example.java:2: error: <identifier> expected
```

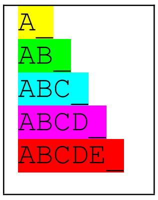
System.out.println(data)

```
17.class Example{
	public static void main(String[] args){
		System.out.println("A");

		System.out.println("B");

		System.out.println("C");

		System.out.println("D");

		System.out.println("E");
	}
}
OUTPUT:===

A
B
C
D
E
```



System.out.print(data)

```
18.class Example{
	public static void main(String[] args){
		System.out.print("A");

		System.out.print("B");

		System.out.print("C");

		System.out.print("D");

		System.out.print("E");
	}
}
OUTPUT:==

ABCDE
```



System.out.print(data) VS System.out.println(data)

19.

```
class Example{
     public   static void                                main(String  args[] ){
              System.out.print("A");

              System.out.println("B");

              System.out.print("C");

              System.out.print("D");

              System.out.println("E");

                       }
}
OUTPUT:===
AB
CDE
```

A_
AB _
AB C_
AB CD_
AB CDE _

20.

```
class Example{
     public   static void  main(String  args[] ){

System.out.println("A");

System.out.println();

System.out.println("B");

System.out.println();

System.out.println("C");
                  System.out.println();
                  System.out.println("D");
                  System.out.println();
                  System.out.println("E");

                  }
}
OUTPUT:===

A

B

C

D

E
```

A -	A …… -	A …. B -	A …. B …. -	A … B …. C -	A … B …. . C …. -	A … B …. . C …. D -	A … B …. . C …. D …. . -	A … B …. . C …. D …. . E -

If we need a **newline** at the end of the string, we should call the **println() method,** which output a newline character appropriate to your platform. that's all about printing newline on java.

```
21.
class Example{
     public   static void  main(String  args[] ){

System.out.print("A");

System.out.println();

System.out.print("B");

System.out.print("C");
                 System.out.print("D");
                 System.out.println();
                 System.out.print("E");
```

A_	A …… -	A …. B_	A …. BC_	A … BCD_	A … BCD …… _	A … BCD ……. E_

```
OUTPUT:===
A
BCD
E
```

```
22. class Example{
     public static void main(String[]
          System.out.print("A");
          System.out.print();
     }
}
```

Example.java:4: error: no suitable method found for print(no arguments)

Java comments

Line comment   // any code

```
23.
class Example{
     public   static void  main(String  args[] ){
                 System.out.println("A");
                 //System.out.println("B");
                 System.out.println("C");
                 //System.out.println("D");
                 System.out.println("E");

                 }
}
OUTPUT:==
A
```

```
C
E
```

Block comment /*any code*/

```
24.
class Example{
     public   static void  main(String  args[] ){
                    System.out.println("A");
                    /*System.out.println("B");
                    System.out.println("C");
                    System.out.println("D");*/
                    System.out.println("E");

                    }
}
OUTPUT:=
A
E
```

Simple data in java(JAVA LITERALS)

```
24.
class Example{
     public   static void  main(String  args[] ){

      String literals
      ============
      System.out.println("Rashmi");      //Rashmi
               System.out.println("A");     //A

                    Character literals
                     =============
                    System.out.println('B');     //B
                    System.out.println('3');     //3
                    //System.out.println('AB');   //Example.java:7:
error: unclosed character literal

                    Integer literal
                    =================
                    System.out.println(12345);   //12345
                    System.out.println(-12345);  //-12345

                    Floating-point literal
                    =================
                    System.out.println(2.3456);  //2.3456
                    System.out.println(-2.3456);   //-2.3456
                    System.out.println(0.001);    //0.001
                    System.out.println(1e-3);     //0.001
                     System.out.println(1000.0);  //1000.0
                    System.out.println(1e3);    //1000.0
                    Boolean Literal
                    ====================
                    System.out.println(true);   //true
```

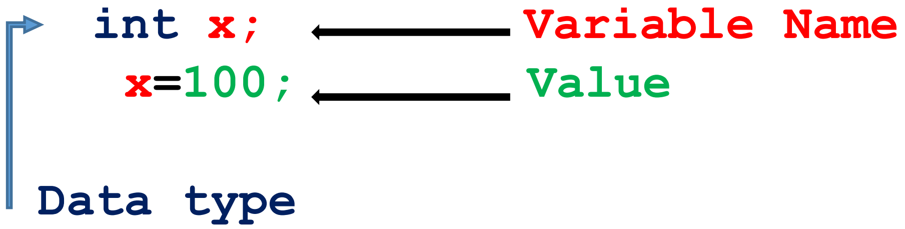
```
                System.out.println(false);   //false
                // System.out.println(True); //Example.java:18:
error: cannot find symbol
                //System.out.println(falsE);  //Example.java:18:
error: cannot find symbol
                }
}
```

	Max value	Min value
String literals	2147483647	0
Character literals	65535	0
Integer literals	2147483647	-2147483647
Floating-point literal	3.40282346638528860+38	1.40129846432481707092372958328991e-45

Computer Variables

Requset memory location for RAM(Random access memory) of temporarily store any data.

(පරිගණක විචල්‍යය යනු ඕනෑම දත්තයක් තාවකාලිකව ගබඩා කිරීම සඳහා (සසම්භාවී ප්‍රවේශ මතකය)ඉල්ලා ගන්න මතක ස්ථානයයි.)



```
int x;
  x=100;
```

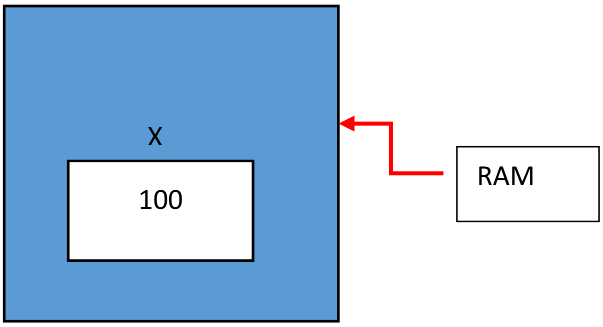
int x; → declare
X=10; → Initiazation
int x=10; → same line declaration and initialization

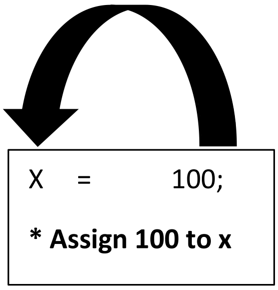
```
int y;
System.out.println(y);    y is not iniazed

Boolean b;
If(b){}   bis not iniazed
```

Dynamic Initialization

int a=2,b=3;
int total=a+b;→ total is dynamically initialized at run time

```
25.
class Example{
public   static void  main(String  args[] ){

 int x; → Varibale declaration and create  a memory location

 x=100; → Assign 100 to x

 System.out.println(x); → Print value of x    // 100
                                            }
}
```





```
26.
class Example{
     public static void main(String[] args){
          int x;
          //x=100;
          System.out.println(x);
           }
}
```

Example.java:5: error: variable x might not have been initialized

Compile error

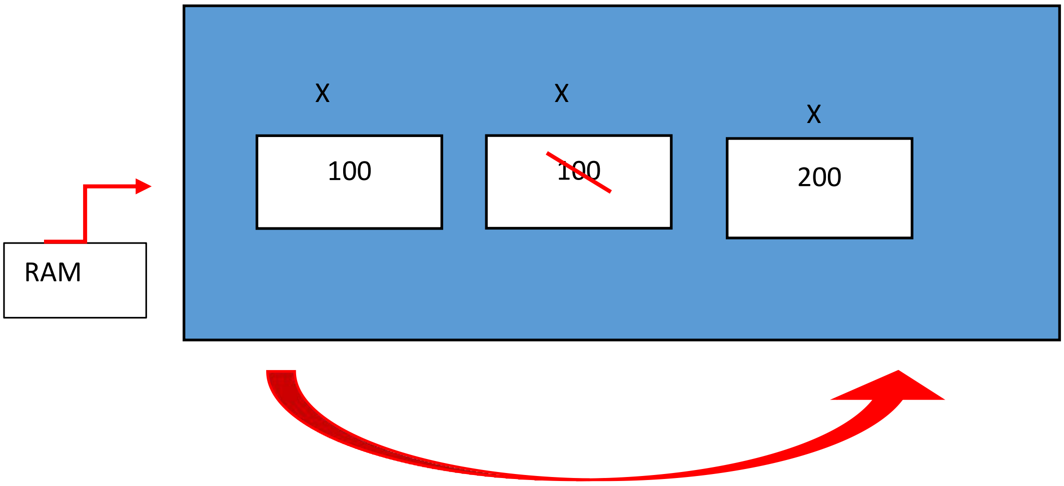
```
27.
class Example{
     public static void main(String[] args){
          int x;
          System.out.println(x);
          x=100;
     }
}
```

Example.java:4: error: variable x might not have been initialized

illegal

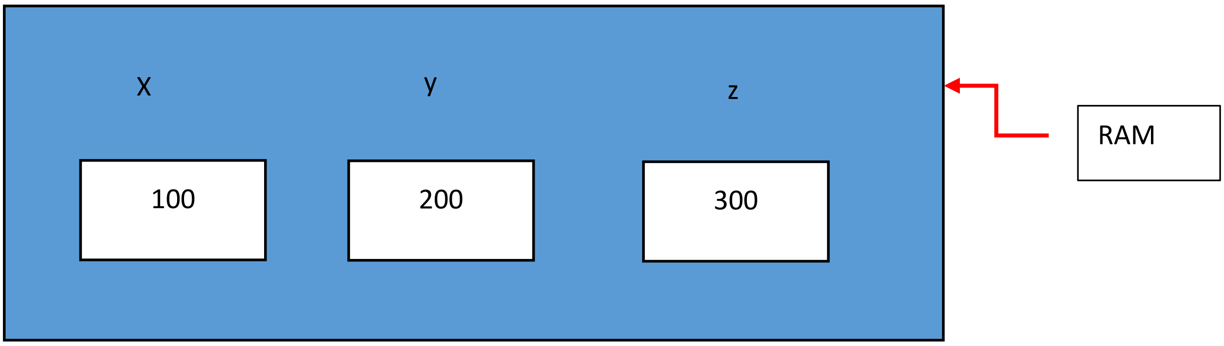
28.

```
class Example{
      public static void main(String[] args){
            int x;
            x=100;
            x=200;
            System.out.println(x); //200
      }
}
```



29.

```
class Example{
       Public static void min(String args[]){
      int x=100;
      System.out.println(x); //100
      int y
      Y=200;
      System.out.println(y);//200
      int z=300;
      System.out.println(z); //300
       }
}
```



leagal

```
30.
int x,y,z
x=100;
y=200;
z=300
    output:-100
           200
```

```
32.
Int x,y,z;
x=y=z=100;
   output:-100
            100
            100
```

```
31.
Int x=100,y,z=300;
y=200;
     output:100
             200
            300
```

Illeagal

```
34.
int x,y,z;
X=y=z=100;
int z;
Z=400;
```

```
35.
Int x=y=z=100;
```

```
36.
class Example{
     public static void main(String[] args){
     System.out.println("10+20+30");        //10+20+30
     System.out.println("10"+"20+30");      //1020+30
     System.out.println("10+20"+"30");      //10+2030
     System.out.println("10"+"20"+"30");    //102030
     System.out.println(10+"20+30");        //1020+30
     System.out.println("10+20"+30);        //10+2030
     System.out.println(10+"20"+"30");      //102030
     System.out.println("10"+20+"30");      //102030
     System.out.println("10"+"20"+30);      //102030
     System.out.println(10+20+"30");        //3030
     System.out.println("10"+20+30);        //102030
     System.out.println(10+20+30);          //60
     }
}
```

```
37.class Example{
     public static void main(String[] args){
          int x=10,y=20,z=30;
          System.out.println("x+y+z");     //x+y+z
          System.out.println("x"+"y+z");   //xy+z
```

```
           System.out.println("x+y"+"z");   //x+yz
           System.out.println("x"+"y"+"z");//xyz
           System.out.println(x+"y+z");     //10y+z
           System.out.println("x+y"+z);     //x+y30
           System.out.println(x+"y"+"z");   //10yz
           System.out.println("x"+y+"z");   //x20y
           System.out.println("x"+"y"+z);   //xy30
           System.out.println(x+y+"z");     //30z
           System.out.println("x"+y+z);     //102030
           System.out.println(x+y+z); //60
      }
}
```

```
38. class Example{
      public static void main(String[] args){
           int x=100,y=200;
           System.out.println(x+" "+y); //100 200
      }
}
```

Keyboard Input

```
import java.util.*;
class  Example{
          public static void main(String args[]){
              Scanner input=new Scanner(System.in);
              System.out.print(“Enter number:-    “);
              int number=input.nextInt();
             }
}
```

Method	Description
nextBoolean()	Reads a boolean value from the user
nextByte()	Reads a byte value from the user
nextDouble()	Reads a double value from the user
nextFloat()	Reads a float value from the user
nextInt()	Reads a int value from the user
nextLine()	Reads a String value from the user
nextLong()	Reads a long value from the user
nextShort()	Reads a short value from the user
next.charAt(0)	Reads a char value from the user

39.

1.Declare 4 variables using only ONE statement. (variable names : Computing ,Maths, Science, English)

int Computing, Maths, Science, English=0;

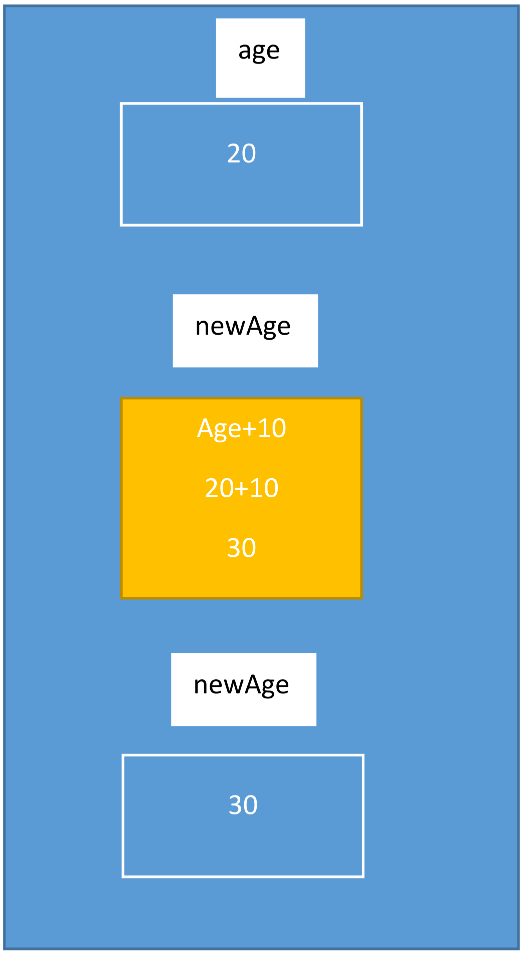
2. Initialize the 4 variables.

```
        System.out.print("Computing:-");
        Computing=input.nextInt();

        System.out.print("Maths:-");
        Maths=input.nextInt();

        System.out.print("Science:-");
        Science=input.nextInt();

        System.out.print("English:-");
        English=input.nextInt();
```

```
40.
int newAge;
newAge=age+10;
age=newAge; //find age after 10 years;
System.out.println("Your age after 10 years : "+age);
```

```
age+=age+10;
```

Shorthand
Assinment



+=

-=

*=

/=

%=

&=

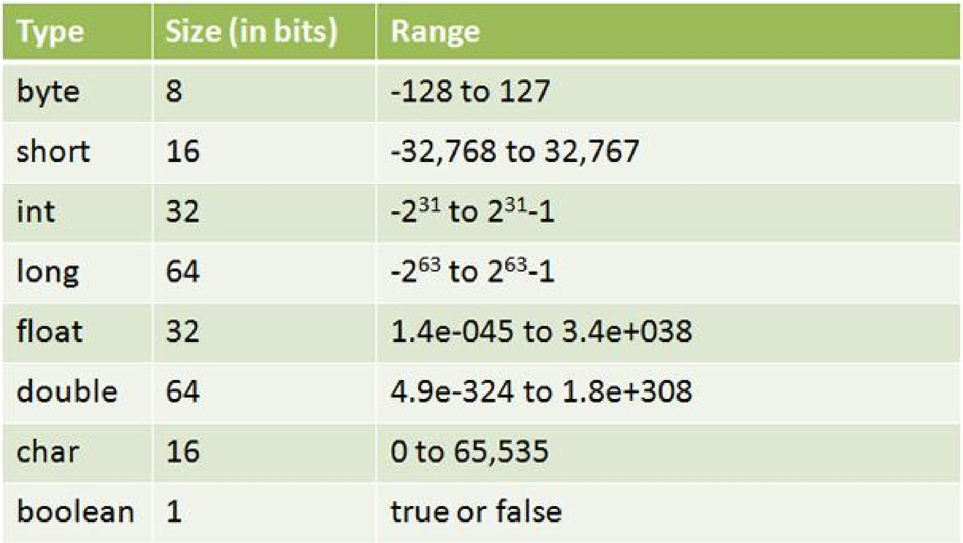
|=

^=

Primitive Data type in java

In Java, the primitive data types are the predefined data types of Java. They specify the size and type of any standard values. Java has 8 primitive data types namely byte, short, int, long, float, double, char and boolean

ජාවා හි, ප්‍රාථමික දත්ත වර්ග යනු ජාවා හි පූර්ව නිශ්චිත දත්ත වර්ග වේ. ඒවා ඕනෑම සම්මත අගයක ප්‍රමාණය සහ වර්ගය නියම කරයි. Java සතුව byte, short, int, long, float, double, char සහ boolean යන ප්‍රාථමික දත්ත වර්ග 8ක් ඇත.



Type	Size (in bits)	Range
byte	8	-128 to 127
short	16	-32,768 to 32,767
int	32	-2^{31} to $2^{31}-1$
long	64	-2^{63} to $2^{63}-1$
float	32	1.4e-045 to 3.4e+038
double	64	4.9e-324 to 1.8e+308
char	16	0 to 65,535
boolean	1	true or false

```
41.class Example{
      public static void main(String[] args){
            int x;
            x=1.5; // Example.java:4: error: incompatible types:
possible lossy conversion from
double to int
            System.out.println(x);
      }
}

42.class Example{
      public static void main(String[] args){
            double x;
            x=1.5; // print 1.5 (double)
            System.out.println(x);
      }
}

43.class Example{
      public static void main(String[] args){
            char x;
            x='A';
            System.out.println(x); //prints A (one character)

            X=”A”;
            System.out.println(x);  // Example.java:6: error:
incompatible types: String cannot be converted to char
            boolean b;
            b=10>9;
```

```
        System.out.println(b); //true
    }
```

Java Literals

A literal is a source code representation of a fixed value. They are represented directly in the code without any computation. Literals can be assigned to any primitive type variable.

<table>
<tr><th>Name of Literals</th><th>Example</th></tr>
<tr><td>Integer Literals</td><td>Class Example{
 Public static void main(String args[]){
   System.out.println(100); //100
   System.out.println(0B1100100); //100
   System.out.println(0b1100100); //100
   System.out.println(0144); //100
   System.out.println(0X64);   //100
   System.out.println(0x64);   //100
  }
}
<table>
<tr><th>Number System</th><th>Base Value</th><th>Numbers and Alphabetic Characters Used</th></tr>
<tr><td><b>Binary</b></td><td>2</td><td>0,1</td></tr>
<tr><td><b>Octal</b></td><td>8</td><td>0,1,2,3,4,5,6,7</td></tr>
<tr><td><b>Decimal</b></td><td>10</td><td>0,1,2,3,4,5,6,7,8,9</td></tr>
<tr><td><b>Hexadecimal</b></td><td>16</td><td>0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F</td></tr>
</table></td></tr>
<tr><td>Floating point Literals</td><td>class Example{
      public static void main(String args[]){
            System.out.println(123.34343); // 123.34343
            System.out.println(1200.0); // 1200.0
            System.out.println(1.2E3); // 1200.0
            System.out.println(.0012); // 0.0012
            System.out.println(1.2e-3); // 0.0012
      }
}</td></tr>
<tr><td>Boolean Literals</td><td>class Example{
      public static void main(String args[]){
            System.out.println(true); //true
            System.out.println(false); //false
            boolean b=10>9;
            System.out.println(b); //true
            System.out.println(6>7); //false
            //System.out.println(<b>truE</b>); //Compile Error
      }
}</td></tr>
<tr><td>String</td><td>class Example{</td></tr>
</table>

Literals	public static void main(String args[]){
	System.out.println("Niroth"); //Niroth
	System.out.println("1245"); //
	1234 System.out.println("1.2E12");
	// 1.21E12
	}
	}

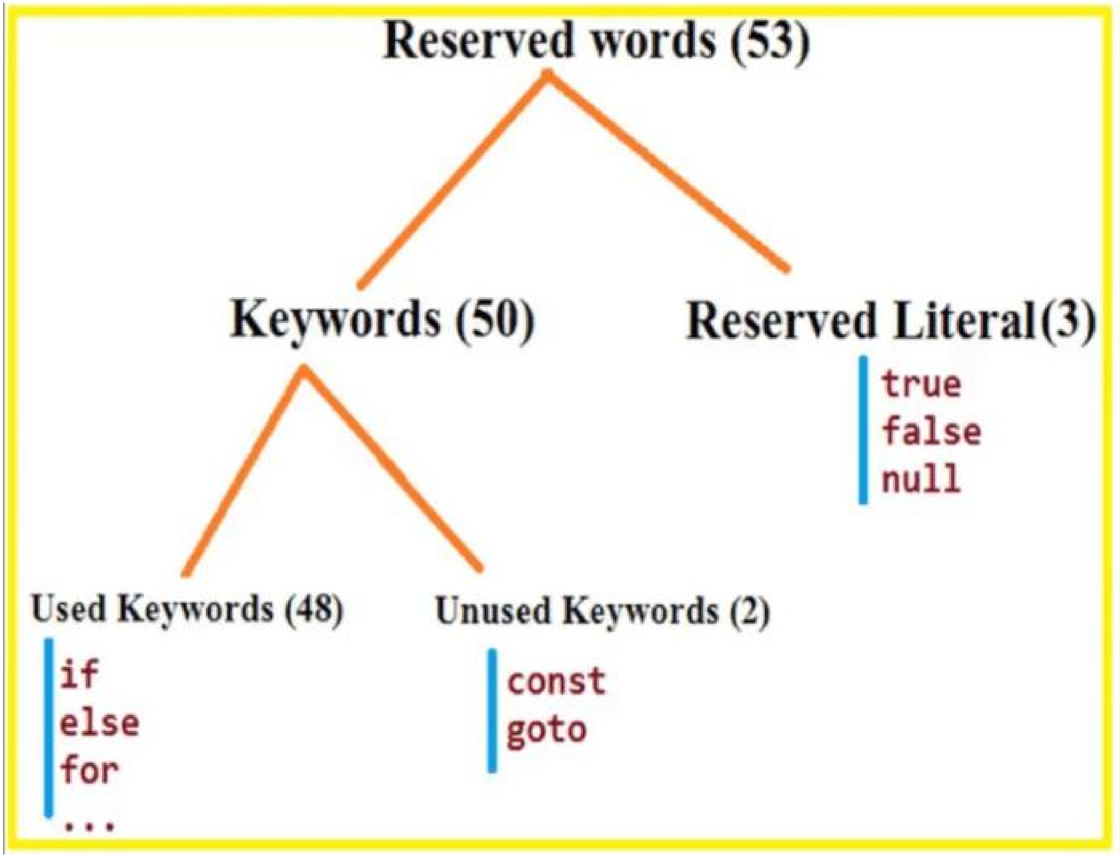
Character Literals (one character)	class Example{
	public static void main(String args[]){
	System.out.println('A'); A
	System.out.println('%'); %
	System.out.println('8'); 8
	System.out.println('**AB**');
	Compiler eror
	}
	}

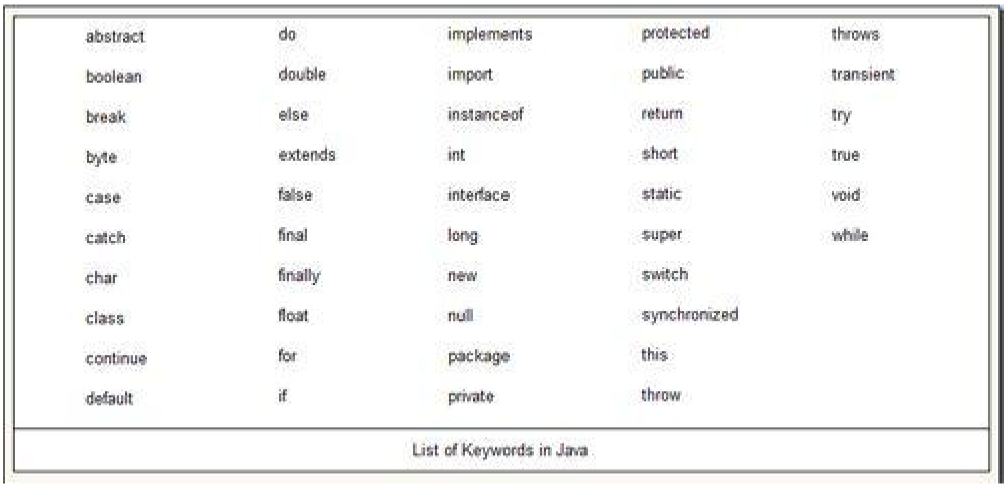
Escape Sequence	**Description**
\t	Insert a tab in the text at this point.
\b	Insert a backspace in the text at this point.
\n	Insert a newline in the text at this point.
\r	Insert a carriage return in the text at this point.
\f	Insert a form feed in the text at this point.
\'	Insert a single quote character in the text at this point.
\"	Insert a double quote character in the text at this point.
\\	Insert a backslash character in the text at this point.

44.

```
class Example{
public static void main(String args[]){
System.out.println("AB"+'\t'+"CD ");  //AB  CD
System.out.println( "AB"+'\b'+"CD ");  //ACD
System.out.println( "AB"+'\n'+"CD ");   //AB
                                       //  CD

System.out.println("AB\bCD"); //AB    CD
System.out.println("AB\tCD");  //ACD
System.out.println("AB\nCD");  //AB
                               //  CD
char ch='  " '; //Legal
System.out.println(ch); //prints "

String s1=" \" ";
System.out.println(s1); print "

System.out.println(" \" "); print "

System.out.println("   C:  \\  Windows  \\  \"  Notepad.exe  \"  ");
C:\Windows\"Notepad.exe"


System.out.println("/ \\ / \\ / \\ / \\ "); print   /\/\/\/\
      }
```

Java Reserved words and keywords





abstract
boolean
break
byte
case
catch
char
class
continue
default
do
double
else
extends
false
final
finally
float
for
if
implements
import
instanceof
int
interface
long
new
null
package
private
protected
public
return
short
static
super
switch
synchronized
this
throw
throws
transient
try
true
void
while

List of Keywords in Java

Java Identifiers

*There should not be any space.
*a-z, A-Z,0-9,$_ The only allowed characters for identifiers are all alphanumeric characters.
*Identifiers shoud not start with digit (0-9) .
*java identifiers are case-sensitive.
*Reserved words can’t be used as a identifier

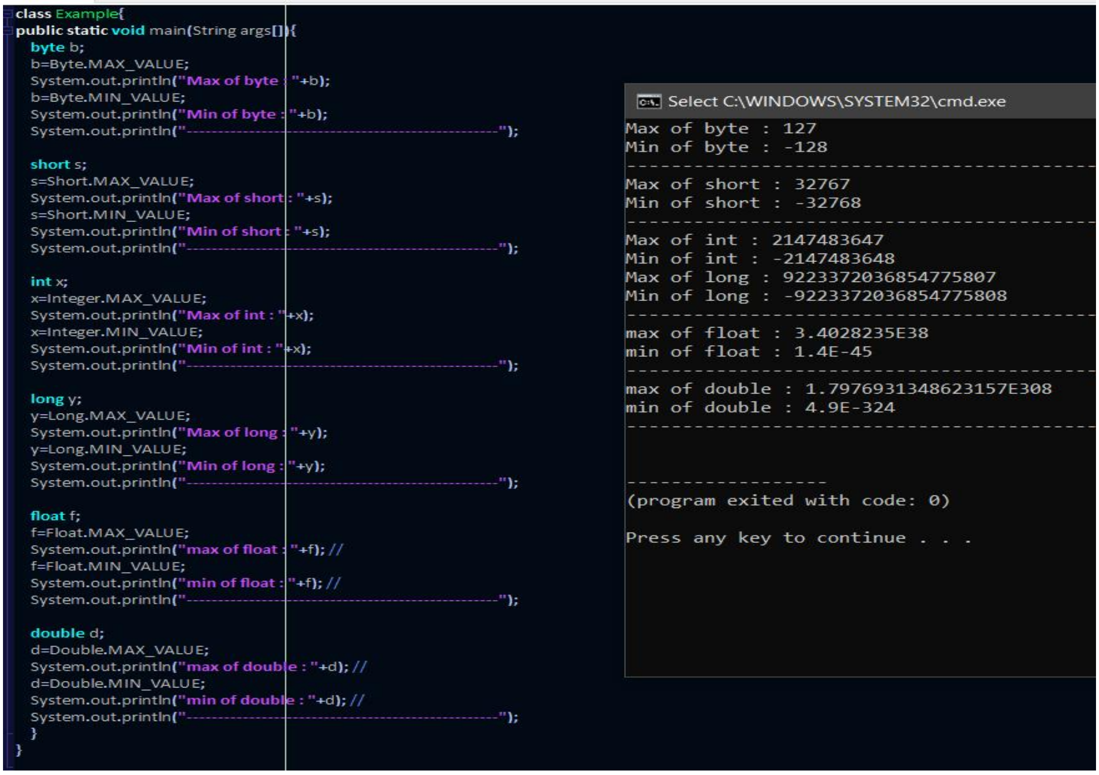
```
    45.class Example{
          public static void main(String args[]){
           //int student Mark;
           //student Mark=0;
           //int 21s;
           //21s=0;
           //Example.java:3: error: not a statement , Example.java:3:
error: ';' expected

           int student_Mark;
           student_Mark=0;

           int s1;
           s1=0;

           int $m;
           $m=0;

           int _mark;
           _mark=0;

           int A;
           a=0; //Example.java:23: error: cannot find symbol

           int while; //Example.java:26: error: illegal start of
expression
           while =0;
        }
}
```

“If the source code file is saved as “Example.java” “

```
46.class example{
    public static void main(String args[]){
    }
}
 // compile ok but runtime error (Error: Could not find or load main
class Example)


47.class xample{
    public static void main(String args[]){
   }
}// compile ok but runtime error (Error: Could not find or load main
class Example)
```

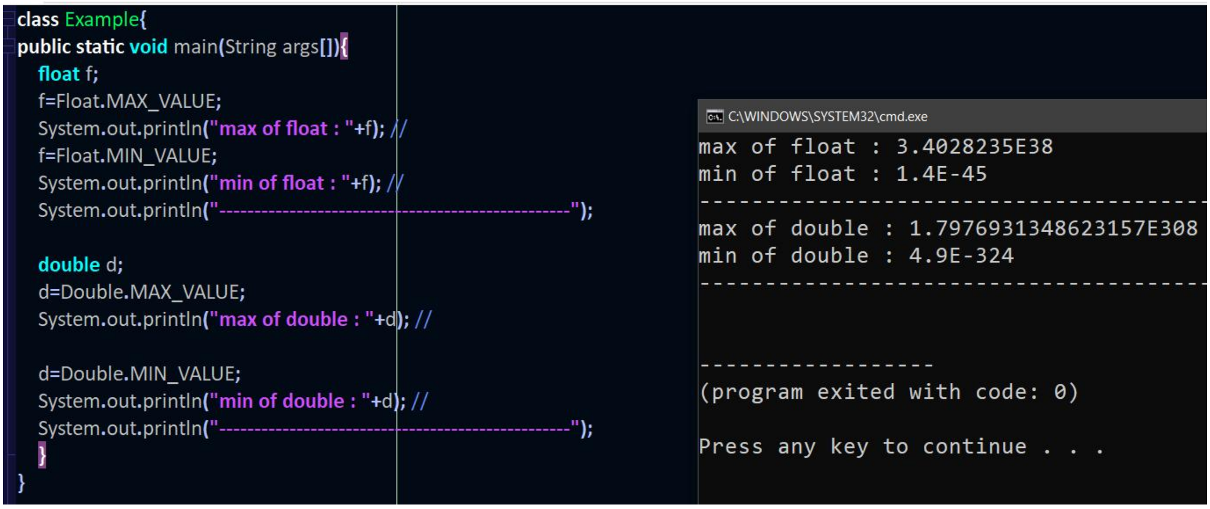
Data Representation in Computer Memory

```
47.
class Example{
	public static void main(String[] args){
		/*int a;
		   a=100;*/

		byte b;
		b=100;
		System.out.println(b); //100
		b=127;
		System.out.println(b); //127

		b=-100;
		System.out.println(b); //-100;

		b=-128;
		System.out.println(b); //-128
	}
}
```



```
class Example{
public static void main(String args[]){
  byte b;
  b=Byte.MAX_VALUE;
  System.out.println("Max of byte : "+b);
  b=Byte.MIN_VALUE;
  System.out.println("Min of byte : "+b);
  System.out.println("----------------------------------------------------");

  short s;
  s=Short.MAX_VALUE;
  System.out.println("Max of short : "+s);
  s=Short.MIN_VALUE;
  System.out.println("Min of short : "+s);
  System.out.println("----------------------------------------------------");

  int x;
  x=Integer.MAX_VALUE;
  System.out.println("Max of int : "+x);
  x=Integer.MIN_VALUE;
  System.out.println("Min of int : "+x);
  System.out.println("----------------------------------------------------");

  long y;
  y=Long.MAX_VALUE;
  System.out.println("Max of long : "+y);
  y=Long.MIN_VALUE;
  System.out.println("Min of long : "+y);
  System.out.println("----------------------------------------------------");

  float f;
  f=Float.MAX_VALUE;
  System.out.println("max of float : "+f); //
  f=Float.MIN_VALUE;
  System.out.println("min of float : "+f); //
  System.out.println("----------------------------------------------------");

  double d;
  d=Double.MAX_VALUE;
  System.out.println("max of double : "+d); //
  d=Double.MIN_VALUE;
  System.out.println("min of double : "+d); //
  System.out.println("----------------------------------------------------");
  }
}
```

```
Max of byte : 127
Min of byte : -128
------------------------------------------
Max of short : 32767
Min of short : -32768
------------------------------------------
Max of int : 2147483647
Min of int : -2147483648
Max of long : 9223372036854775807
Min of long : -9223372036854775808
------------------------------------------
max of float : 3.4028235E38
min of float : 1.4E-45
------------------------------------------
max of double : 1.7976931348623157E308
min of double : 4.9E-324
------------------------------------------



------------------
(program exited with code: 0)

Press any key to continue . . .
```



```
class Example{
public static void main(String args[]){
   float f;
   f=Float.MAX_VALUE;
   System.out.println("max of float : "+f); //
   f=Float.MIN_VALUE;
   System.out.println("min of float : "+f); //
   System.out.println("-----------------------------------------------");

   double d;
   d=Double.MAX_VALUE;
   System.out.println("max of double : "+d); //

   d=Double.MIN_VALUE;
   System.out.println("min of double : "+d); //
   System.out.println("-----------------------------------------------");
   }
}
```

```
max of float : 3.4028235E38
min of float : 1.4E-45
-----------------------------------------------
max of double : 1.7976931348623157E308
min of double : 4.9E-324
-----------------------------------------------


------------------
(program exited with code: 0)

Press any key to continue . . .
```

**floatවලයි doubleවලයි min number එක ධන සංඛ්‍ය වීමට හේතුවන්නේ එම කුඩම දශම සංඛ්‍ය නිසයි. 1.45E-45 යනු "0" 45කට පසු 1.45 තිබිමයි*
f>0=ture

48.

```
class Example{
public static void main(String args[]){
        byte b;
         b=Byte.max_value; //Example.java:4: error: cannot find
symbol     symbol:   variable max_value
         System.out.println("Max of byte : "+b);}
     }
```

49.

```
class Example{
public static void main(String args[]){
    System.out.println(2147483647); //max of int(32bits)
    System.out.println(-2147483648);//min of int(32bits)
    System.out.println(2147483648); //max+1  integer number too
large: 2147483648
    System.out.println(-2147483649);//min-1   integer number too
large: -2147483649
     }
}
```

50.

```
class Example{
public static void main(String args[]){
        System.out.println(2147483647);     //max of int(32bits)
        System.out.println(-2147483648);    //min of int(32bits)
         System.out.println(2147483648L);   //Legal, l or L-->64bits
2147483648
         System.out.println(-2147483649L); //Legal, l or L-->64bits
-2147483649
```

```
      }
}
```

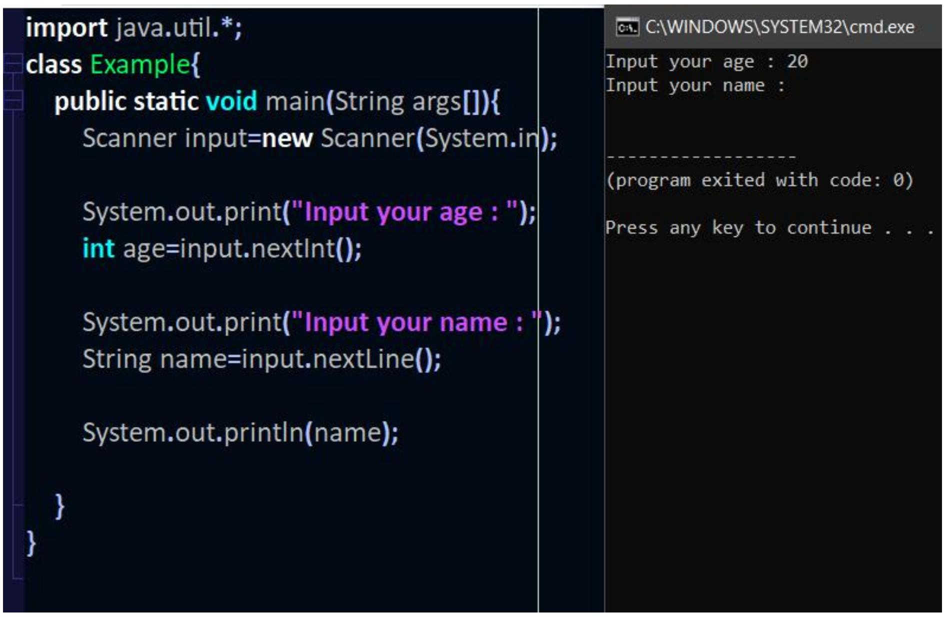
51.

```
class Example{
public static void main(String args[]){
       System.out.println(9223372036854775807L);  //Long.MAX_VALUE
9223372036854775807
      System.out.println(-9223372036854775808L);  //Long.MIN_VALUE
-9223372036854775808
      //System.out.println(9223372036854775808); //max of long+1
Example.java:6: error: integer number too large: 9223372036854775808
      System.out.println(9223372036854775808f);   //Legal, f-->32bits
float   9.223372E18
      System.out.println(9223372036854775808D);   //Legal, d-->64bits
double
9.223372036854776E18
      }
}
```

52.

```
class Example{
      public static void main(String args[]){
            char ch='A';
            System.out.println(ch); //print A

            ch=66;
            System.out.println(ch); //print B
            System.out.println(ch+100);  //165    "char"a  is
numerical  data type
            System.out.println(ch+"1");   //A1
```

53



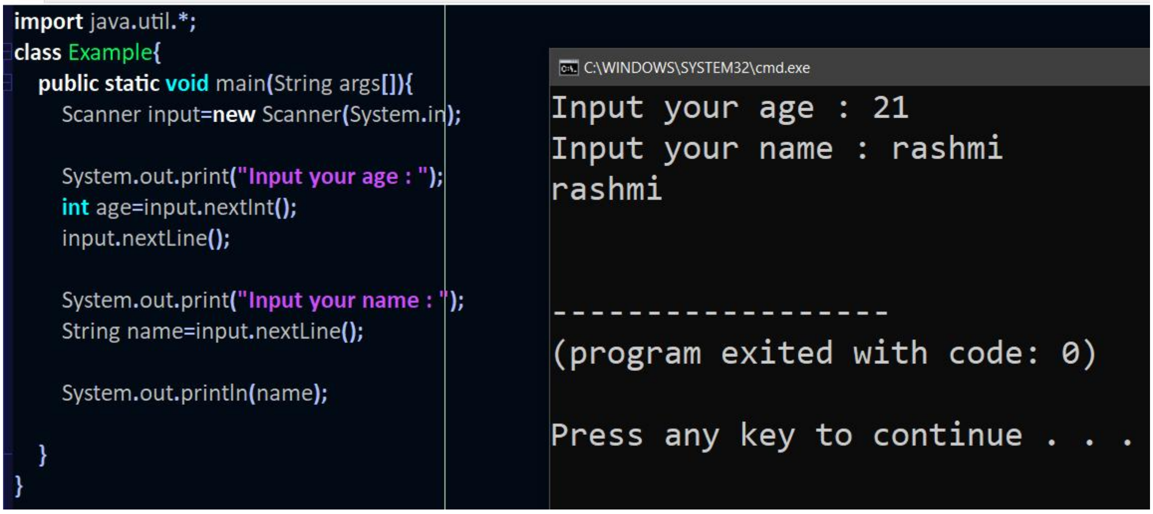
```
import java.util.*;
class Example{
  public static void main(String args[]){
    Scanner input=new Scanner(System.in);

    System.out.print("Input your age : ");
    int age=input.nextInt();

    System.out.print("Input your name : ");
    String name=input.nextLine();

    System.out.println(name);

  }
}
```

```
Input your age : 20
Input your name :


------------------
(program exited with code: 0)

Press any key to continue . . .
```

.

Name එක print නොවීමට හේතුව වන්නේ 20ට පස්සේ enter key එක දෙන නිසා

54.



```
import java.util.*;
class Example{
	public static void main(String args[]){
		Scanner input=new Scanner(System.in);

		System.out.print("Input your age : ");
		int age=input.nextInt();
		input.nextLine();

		System.out.print("Input your name : ");
		String name=input.nextLine();

		System.out.println(name);

	}
}
```

C:\WINDOWS\SYSTEM32\cmd.exe

```
Input your age : 21
Input your name : rashmi
rashmi


------------------
(program exited with code: 0)

Press any key to continue . . .
```

එ error එක නැති කිරීමට "input.nextLine();" යොදයි.

Data Types and bit size

byte 8bit	float 32bit	double 64bit	boolean 1bit
char 16bit	long 64bit	int 32bit	shout 16bit

Signed bit

+65=01000001

-65=11000001

ASCII-American Standard Code for Information Interchange.

A=65	a=97	0=48
B=66	b=98	1=49
C=67…..	c=99…..	2=50

Binary Values(8bit)

128	64	32	16	8	4	2	1
1	1	1	1	1	1	1	1

1 s`Compliment

+10=

0	0	0	0	1	0	1	0

1 s` com:-

1	1	1	1	0	1	0	1

1→0
0→1

2 s`Compliment

+10=

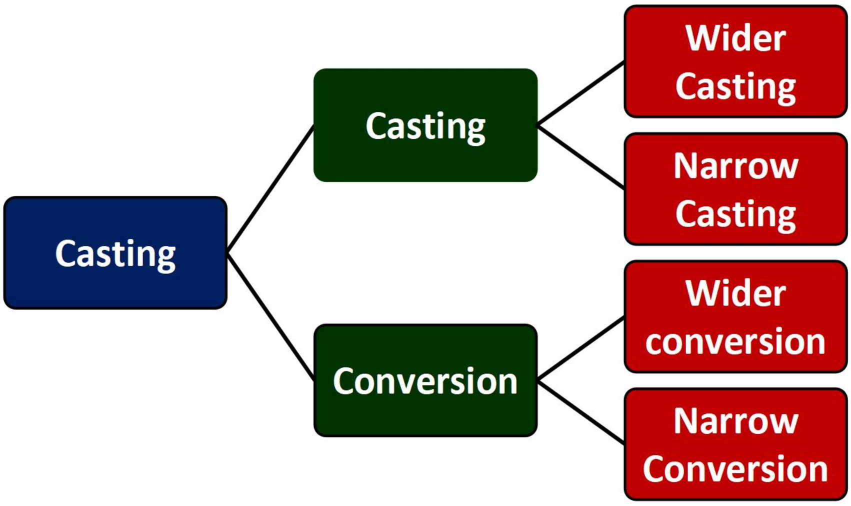
0	0	0	0	1	0	1	0

1 s` com:-

1	1	1	1	0	1	0	1	+

2 s` com:-

	1	1	1	1	0	1	0	1
							+	1
-10	1	1	1	1	0	1	1	0



Data Types Conversion and Casting

Conversion

```
import java.util.*;
class Example{
      public static void main(String args[]){
        byte b=10;
        short s;
        s=b; //Legal, Conversion
        System.out.println(b+" "+s); //10 10
       }
}
```

Byte b=10 covert to binary number
10→00001010_2

8bit

0	0	0	0	1	0	1	0

16bit

0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0

short s=10 covert to binary number

Casting

```
import java.util.*;
class Example{
    public static void main(String args[]){
     short s=10;
    byte b;
    //b=s; //Illegal, incompatible typs
    b=(byte)s;
    //casting, assign last 8bite to b
    System.out.println(b+" "+s); //10 10
       }
}
```

16bit

0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0

Get the last digits

8bit

0	0	0	0	1	0	1	0

```
import java.util.*;
class Example{
     public static void main(String args[]){
          short s=128;
          byte b;
          b=(byte)s; //casting, assign last 8bits of s to b
          System.out.println(b+" "+s); //-128 128
     }
}
```

Short s=128 -→0000000010000000$_2$

0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0

16bit

1	0	0	0	0	0	0	0

8bit

-128

1	0	0	0	0	0	0	0
0	1	1	1	1	1	1	1
						+	1
1	0	0	0	0	0	0	0

```
s=255;
b=(byte)s; //casting, assign last 8bits of s to b
System.out.println(b+" "+s); //-1 255
```

Short s=128 -→0000000011111111$_2$

0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1

16bit

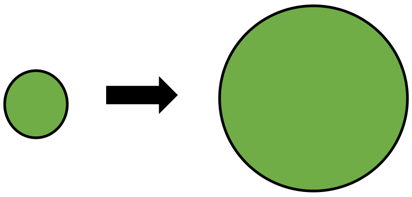
1	1	1	1	1	1	1	1

8bit

-1

0	0	0	0	0	0	0	1
1	1	1	1	1	1	1	0
						+	1
1	1	1	1	1	1	1	1

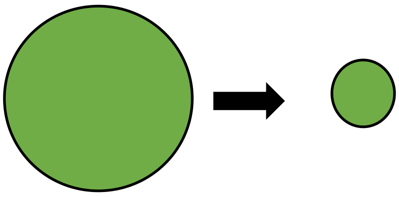
Wider conversion



Automatically.

```
class Example{
public static void main(String[]
args){
        byte b=100;
        short s;
        s=b;
        System.out.println(s+" "+b);
// 100   100
        char ch='A';
        double d=0.12345;
        System.out.println(d+ch);
//65.12345  (65+0.12345)

        }
}
```

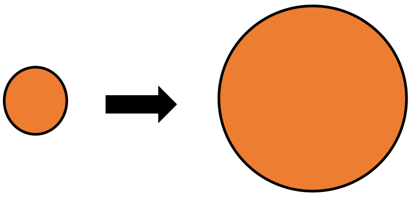
Narrow Conversion



Automatically.

```
class Example{
public static void main(String[]
args){
        int age=20;
        long a=10;
        age=age+a; //Illegal
incompatible types: possible lossy
conversion from long to int
        age+=a;
        System.out.println(age); //30

        }
}
```

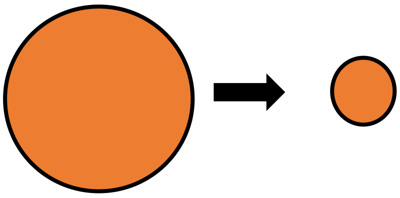
Wider Casting



Not automatically.

```
class Example{
public static void main(String[]
args){
        char ch='A';
        System.out.println(ch);
//prints A
        System.out.println((int)ch);
//prints 65-->wider casting
        int x=123;
        System.out.println(x);
//prints 123
        System.out.println((double)x)
; //prints 123.0
        int a=5,b=2;
        System.out.println(a/b);
//2,-->integer division

System.out.println((double)a/b);
//2.5-->floating-point division
        System.out.println(a/(double)
b); //2.5-->floating-point division
        }
```

Narrow Casting

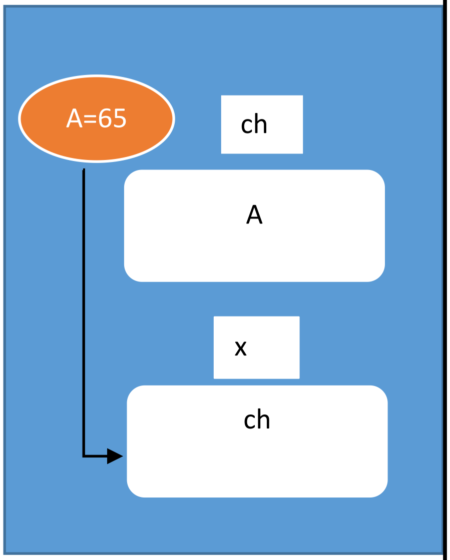


Not automatically.

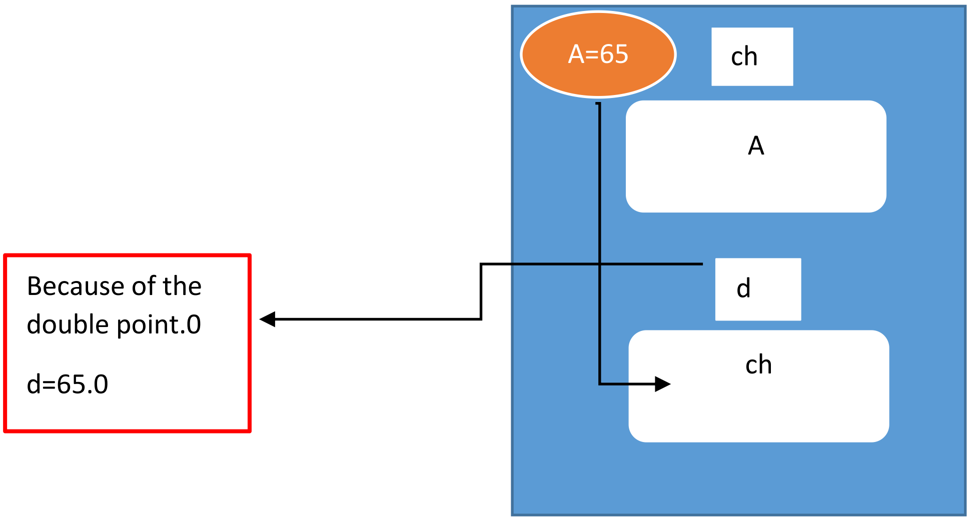
```
class Example{
public static void main(String[]
args){
        short s=100;
        byte b;
        b=(byte)s;
        System.out.println(b+" "+s);

        char ch='A';
        double d=1.12345;
        System.out.println(d+ch);
//66.12345
        System.out.println((int)d+ch)
; //6
        double x=1.12345;
        double y;
        y=x+ch;
        System.out.println(y);
//66.12345
        y=(int)x+ch;
        System.out.println(y); //66.0
}
}
```

}	

```
class Example{
	public static void main(String args[]){
		int x=100;
		short s;
		long y=10;
		float f=10;
		double d=10;

		s=x; //Example.java:9: error: incompatible types: possible lossy conversion from int to short
		x=y; //Example.java:10: error: incompatible types: possible lossy conversion from long to int
		y=f; //Example.java:11: error: incompatible types: possible lossy conversion from float to long
		f=d; //Example.java:12: error: incompatible types: possible lossy conversion from double to float

		d=s;
		d=x;
		d=y;
		d=f;

		f=s;
		f=x;
		f=y;

		y=s;
		y=x;

		x=s;

		System.out.println();
	}
}
class Example{
	public static void main(String[] args){
		char ch='A';
		int x;
		x=ch; //Legal
		System.out.println(ch+" "+x); //A 65
	}
}
```



```
class Example{
     public static void main(String[] args){
          char ch='A';
          double d;
          d=ch; //Legal
          System.out.println(ch+" "+d); //A 65.0
     }
}
```



Because of the double point.0

d=65.0

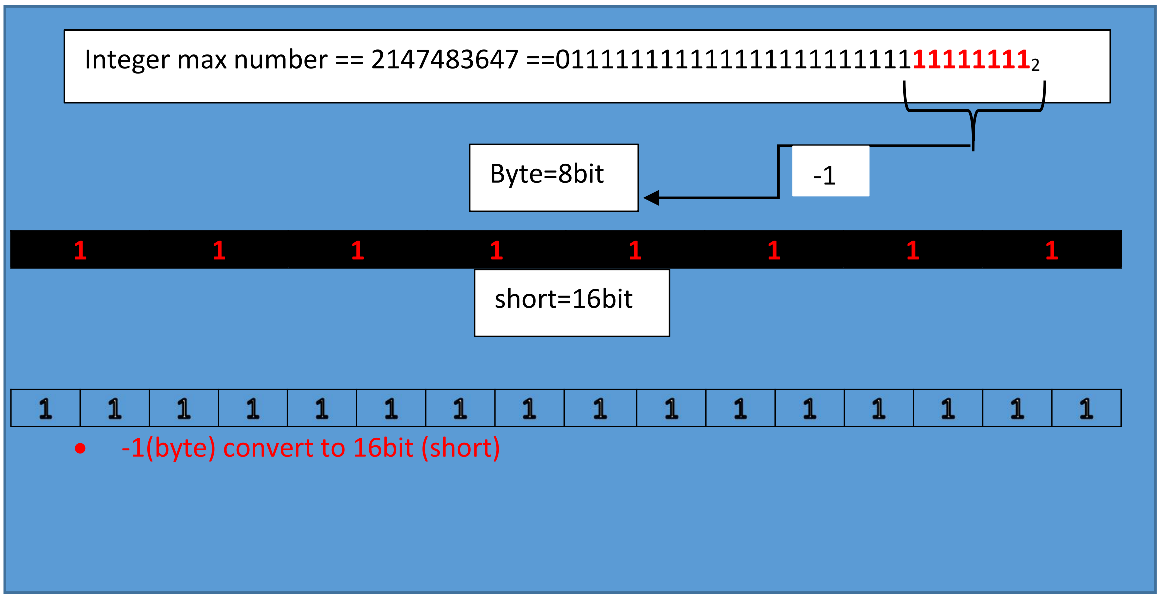
```
class Example{
public static void main(String[] args){
      char ch='A';
      int x=100;
      short s=100;
      byte b=100;

      ch=x; //Illegal incompatible types: possible lossy conversion
from int to char
      ch=b;//Illegal  incompatible types: possible lossy conversion
from byte to char
      ch=s;//Illegal  incompatible types: possible lossy conversion
from short to char

      x=ch; //Legal
      s=ch; //Illegal incompatible types: possible lossy conversion
from char to short
      b=ch; //Illegal incompatible types: possible lossy conversion
from char to byte
      }
}
```

Byte>short > long>float>double

⬆

Char---0-65535

```
class Example{
      public static void main(String args[]){
            int x=66;
            char ch;
            //ch=x; //Illegal
            ch=(char)x;
            System.out.println(ch+" : "+x); //B : 66

            double d=67.12345;
            //ch=d; //Illegal
            ch=(char)d;
            System.out.println(ch+" : "+d); //C : 67.12345

      }
}

class Example{
      public static void main(String[] args){
            int x=Integer.MAX_VALUE;
            System.out.println(x);      2147483647

            short s;
            s=(byte)x;
            System.out.println(s);      -1
      }
}
```



Integer max number == 2147483647 ==$01111111111111111111111111111111_2$

- -1(byte) convert to 16bit (short)

End...