

## Creating New file

JavaBasics.java

[Java extension for java file]

1. define class

Public class JavaBasic

↳ Name of the file (exact name)  
Class name & file name should be exactly same

2. define function inside class

Public static void main (String args []) {  
}

Functr → magic button → Name - Main

↓  
1st thing that java compiler  
searches while compiling

String

↳ S →

capital.

## BoilerPlate code

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

## Output in Java

System.out.print ("Hello World");  
↓                    output  
Function

( ) → parenthesis  
" " → double quote

; → semi colon  
↓

statement  
terminator

S → capital letter  
O → small letter  
P → small letter

## File - Firstcode.java

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

System.out.println ("Hello World");  
System.out.println ("Hello World");

↓  
In (Next line)

Print  
println

System.out.print ("Hello World n");  
                  ↓  
                  Next line

## Output in Java

1. print      System.out.print ("      ");
2. println     System.out.println ("      ");
3. "\n"        System.out.print ("      \n");

" " → string ↴  
Paragraph

## Print Pattern

Pattern : \* \* \* \*

\* \* \*

\* \*

\*

→ (SISO) → short form

System.out.println ("\* \* \* \*");  
SISO ("\* \* \*");  
SISO ("\* \*");  
SISO ("\*");

## Variables in Java



a = 5 , b = 10

2 \* (a + b)  
  ↓            ↓  
literal       variable (values can change)

↓  
(Values are  
universal, doesn't change)

data type    int a = 10 ;    → //variable defined  
                ↓            ↓  
                Name       value

→ assigned

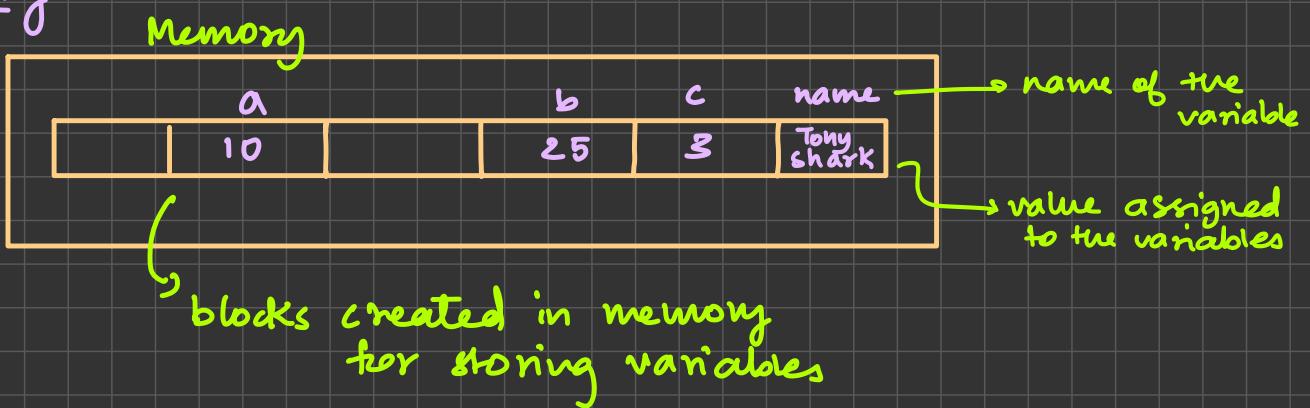
variable names - a,b,c

classes name - javabasic

function name - main, print, println

} all the names  
in Java  
Identifiers

## Memory



- Each block have its own address
- Size of each block may differ.

File name - variable.java

```
public class variable {  
    public static void main (String args []) {  
        int a = 2;  
        int b = 5;  
        System.out.println (a);  
        System.out.println (b);  
  
        String name = "Tony Shark";  
        System.out.println (name);  
  
        a = 50;  
        System.out.println (a);  
    }  
}
```

→ no need to write inside ""  
it will print the exact thing inside it

→ string = S capital =

→ update the value of the variable

## Data types in Java

### Primitive D.T

Byte  
short  
char  
boolean  
int  
long  
float  
double

### Non-Primitive D.T

String  
ArrayList  
Class  
Object  
Interface

Java, C++, Python → Typed language

all data type start with small letter.

Byte (upto 256 only) —

- not used much

char (only one character, all are available)

Boolean (true, false)

float (decimal)

int (whole value)

long (long value int)

double (long value decimal)

short (Number) —

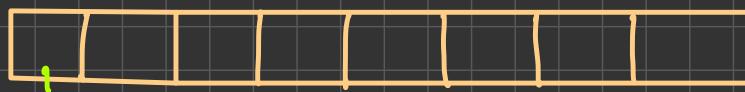
size of Data type (According to size, stored in memory)

Bytes (8 bit)

single information



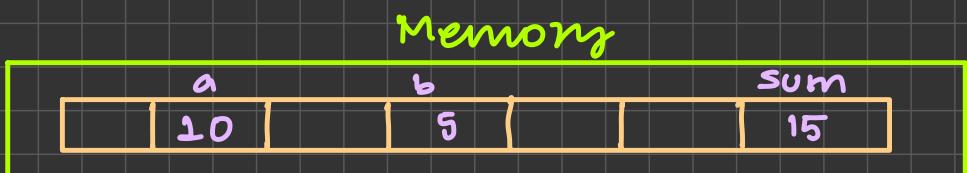
Byte - 8 bit combination



	<u>size</u>	<u>Range</u>	<u>Total</u>
Byte	1 byte	[-128 To 127]	256
short	2 "	[ ]	→ article!
char	2 "	[a, b, c, ..., z] [A, ..., Z] [@, /, ..]	
Boolean	1 "	[true, false]	
float	4 "	[ ]	
int	4 "	[-2 Billion To + 2 Billion]	
long	8 "		
double	8 "		

## Sum of two numbers

```
int a = 10  
int b = 5  
  
int sum = a + b  
System.out.println(sum);
```



## Comments in Java

Statement written in English for understanding.  
It is not executed.

```
// → Single line comments  
/* → Multiline comments */
```

## Input in Java

```
import java.util.*;
```

```
public class Input {  
    public static void main (String args []) {  
        Scanner sc = new Scanner (System.in);  
        String input = sc.next();  
        System.out.println (input);  
    }  
}
```

*data type* → *Scanner object Name*  
*Name of the variable* → stores input upto space

```
String input = sc.nextLine();
```

```
int number = sc.nextInt();
```

To take input as number

can be used to scan whole line  
(name + surname)

next → to store only one word

nextLine → To store whole sentence

sc.nextInt → To store numbers

sc.nextFloat → " " float

Capital

## Inputs in Java

```
next
nextLine
nextInt
nextByte
nextFloat
nextDouble
nextBoolean
nextShort
nextLong
```

}

## Stack Overflow



Solutions to errors & problems

## Sum of a & b

```
import java.util.*;
```

```
public class sumofab {
    public static void main (String args [ ]) {
        Scanner sc = new Scanner (System.in);
        int a = sc.nextInt ();
        int b = sc.nextInt ();
        int sum = a + b;
        System.out (sum);
    }
}
```

## Product of two numbers

```
import java.util.*;
```

```
public class productoftwo {
    public static void main (String args [ ]) {
        Scanner sc = new Scanner (System.in);
        int a = sc.nextInt ();
        int b = sc.nextInt ();
        int product = a * b;
        System.out.print (product);
    }
}
```

## Area of Circle

read.

$$\text{area} = \pi * \text{read} * \text{read}$$

$\downarrow$

3.14 or  $22/7$

float area.  $\rightarrow$  float

Pre-define 3.14 as float  
or else java considers  
it as double value.

## Type conversion in Java      widening/implicit conversion

Conversion happens when :

(1) type compatible

(2) destination type > source type

↳ byte < short < int < float < long < double

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

int a = 25;

long b = a;

System.out.println (b);

→ Possible

b = 25

/\* long a = 25;

int b = a;

System.out.print(b); \*/

→ Not possible

: 'long' > 'int'

Condition (2) not satisfied

}

```
import java.util.*;
```

```
public class typeconversion {
```

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

```
        Scanner sc = new Scanner (System.in);
```

int num = sc.nextFloat();

System.out.println (num);

→ Not possible

condtn (2) - not satisfied

float num = sc.nextInt();

System.out.print (num);

→ 15  
15.0

}

## Type Casting in Java

narrowing / explicit conversion

float a = 25.0;

int b = a; X

int b = (int) a; ✓

```

public class typecasting {
    public static void main (String args []) {
        float a = 25.12;
        int b = (int) a;
        System.out.println (b);
    }
}

```

```

import java.util.*;
public class typecasting1 {
    public static void main (String args []) {
        Scanner sc = new Scanner (System.in);
        float marks = 99.999f;
        int marks2 = (int) marks;
        System.out.println (marks2);
    }
}

```

It can be used to convert character into numeric values.

```

public class typecasting1 {
    public static void main (String args []) {
        char ch = 'a';
        char ch2 = 'b';
        int number = ch;
        int number2 = ch2;
        System.out.println (number);
        System.out.println (number2);
    }
}

```

→ 97  
98

## Type Promotion in Java

$$\begin{array}{ccccccc}
 & a + b * c / e & & & & & - \text{operation between} \\
 & / \quad | \quad | \quad \backslash & & & & & \text{diff. data types} \\
 \text{int} & \text{float} & \text{char} & \text{long} & & &
 \end{array}$$

- Java automatically promotes each byte, short or char operand to int when evaluating an expression
- If one operand is long, float or double the whole expression is promoted to long, float, or double respectively. (convert to largest data type present in the exp.)

```
{
    char a = 'a';
    char b = 'b';
    char c = a - b;
    System.out.println((int)(b));
}
```

$a - b \rightarrow$  mathematical exp.  
 $\downarrow$   
 conv. to int

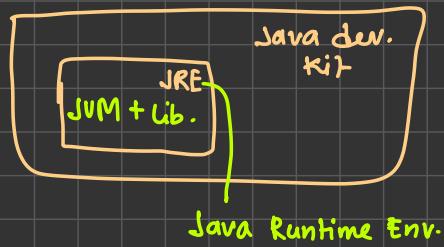
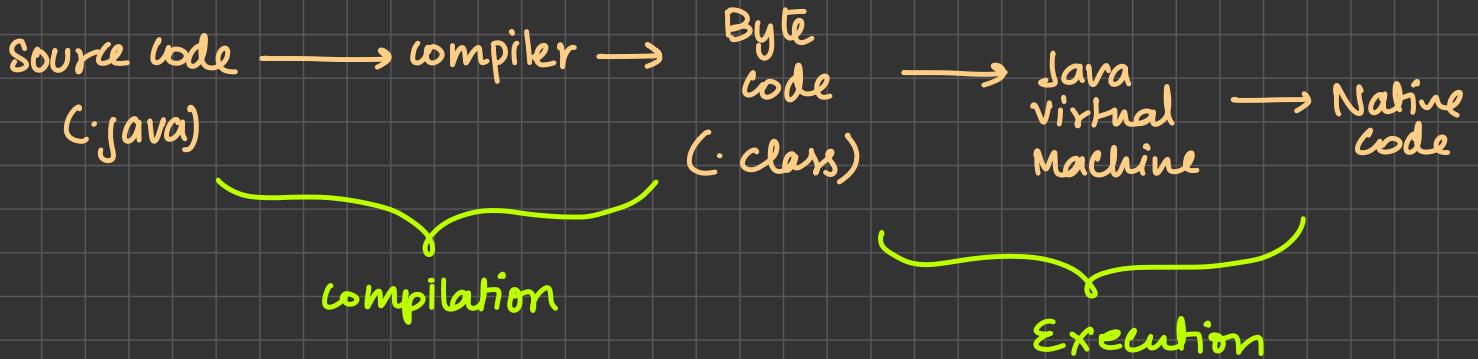
```
{
    short a = 5;
    byte b = 25;
    char c = 'c';
    byte bt = a + b + c;
}
```

$\hookrightarrow$  can't change int to byte  
 $\text{byte bt} = \text{byte}(a + b + c)$   
 (type casting)

$\text{byte b} = 5;$ Wrong $\rightarrow b = \underline{b * 2};$ $\checkmark b = (\text{byte}) b * 2;$	$\rightarrow$ operator $\rightarrow$ so, b converted into int & then mult. $\downarrow$ can't store in b
---	--

↓  
type promotion

## How is our code Running?



## Practise Questions

Q1. In a program , input 3 numbers : A, B, C . You have to output the average of these 3 numbers .

File name - Average of three number.java

```
Public class AverageofthreeNumber {  
    public static void main (String args []) {  
  
        int a = 5;  
        int b = 7;  
        int c = 18;  
  
        int average = (a+b+c) / 3;  
  
        System.out.println (average);  
    }  
}
```

or,

```
import java.util.*;  
  
public class averageofthreenumbers {  
    public static void main (String args []) {  
        Scanner sc = new Scanner (System.in);  
  
        int a = sc.nextInt();  
        int b = sc.nextInt();  
        int c = sc.nextInt();  
  
        int average = (a+b+c) / 3;  
  
        System.out.print (average);  
    }  
}
```

↳ System.out.println ("average is :" + average);

### Pseudocode

1. Input a, b, c
2. Store the average formula
3. Now output the average.

Q2. In a program, input the side of a square . You have to output the area of the square.

file name: areaofsquare.java

```
import java.util.*;  
  
public class areaofsquare{  
    public static void main (String args []){  
        Scanner sc = new Scanner (System.in);  
  
        int side = sc.nextInt();  
  
        int area = side * side;  
  
        System.out.println (area);  
    }  
}
```

Q3. Enter cost of 3 items from the user (using float data type) - pen, pencil and eraser.

You have to output the total cost of the items back to the user as their bill

(add on: you can also try adding 18% gst tax on the items in the bill)

file name: bill.java

```
import java.util.*;  
  
public class bill {  
    public static void main{  
        Scanner sc = new Scanner (System.in);  
  
        float pen = sc.nextFloat ();  
        float pencil = sc.nextFloat ();  
        float eraser = sc.nextFloat ();  
  
        float bill = float (pen + pencil + eraser);  
        System.out.println (bill);  
    }  
}
```

→ error  
(.: operation → all D.T  
are  
converted  
into int  
↓  
Use Type Casting

Q4. What will be the type of result in the following Java code?

```
byte b = 4;  
char c = 'a';  
short s = 512;  
int i = 1000;  
float f = 3.14f;  
double d = 0.99954;
```

→ floats are always written this way, cause Java take decimal value as double if it's not pre-defined.



Type promotion

→ double (∴ it's the highest data type)