5/08/22

VARIABLES AND DATATYPES

<u>boilerplate code</u>: code which is written every time

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

<u>O/P in Java</u>

System.out.(print) ("Hello World!"); ← terminator

O/P

func^

To run :- cd path-of-file

Javac file_name for compilation

Java file_name for running

* println(" ") => after print statement give space OR \n

<u>Print pattern</u>

```
****
***
**
*
```

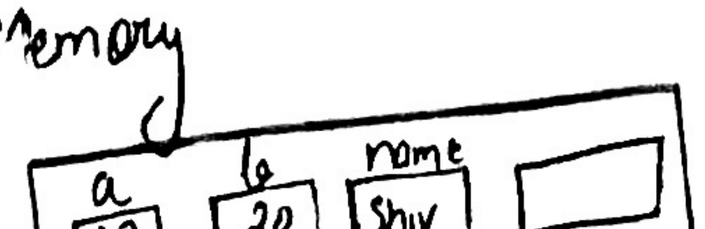
Variables in Java

Literal: Does not change . eg: 2, 3, 'a', 'b'

Variables : can vary . eg- a, b, c, ...

identifier : func^ name, class name, variable name

Memory



code :

```
int a=10          => 10
S.o.p (a)
```

Data Types in java

- Java is typed language : Tell what type of data you want to use

Data Types

primitive	Non-primitive
• byte 1b [-128 to 127]	String
• short 2b	Array
• char 2b ['a' to 'z' ---]	Class
• boolean 1b [true, false]	object
• int 4b [-2 Billion to +2 Billion]	Interface
• long 8b for long int	
• float 4b decimal	
• double 8b for long decimal	

b → bit

Comments in Java

Single line comment : //

Multi line comment : /* code statement */

Input in Java

next nextFloat nextLong
nextLine next Double
nextInt nextBoolean
nextByte nextShort

import java.util.*;
// Boilerplate code

```
         class   object              constructor
        Scanner sc = new Scanner (System.in);
dtype ↘     ↙ var name
→ String input = sc.next();                One word
  S.o.p (input)

→ String name = sc.nextLine();              multiple line
  S.o.p (name)

→ int number = sc.nextInt();                number
  S.o.p (number);

→ float price = sc.nextFloat();             float
  S.op (price)
```

* 3.14 ⇒ ~~float~~ double (any decimal)
3.14f ⇒ float

Renu b

Type Conversion / widening conversion / implicit conversion

Conversion happens when

a) type compatible   b) destination type > source type

byte > short > int > float > long > double

Type casting / narrowing conversion / explicit conversion

- forceful conversion

  int b = (int) a   ← want to cast

- 'a' → 97

Type promotion in Expression

- Only applicable in expression

1) Java automatically promotes each byte, short, or char operand to int when evaluating an expression.

2) If one operand is long, float or double the whole expression is promoted to long, float, or double respectively.

Eg: byte b = 5
    b = b * 2; ✗
    (byte ← b; b → int; 2 → int)

byte b = 5;
b = (byte)(b * 2);

How is our code running

Source code (.java) → compiler → Byte Code (.class) → JVM → Native code (0,1)

Compilation: Source code → compiler → Byte Code

Execution: JVM → Native code

Renu b

- Portable : Native code adjust acc. to OS