

15

February

8th Wk • 046-319

Tuesday

~~JAVA~~

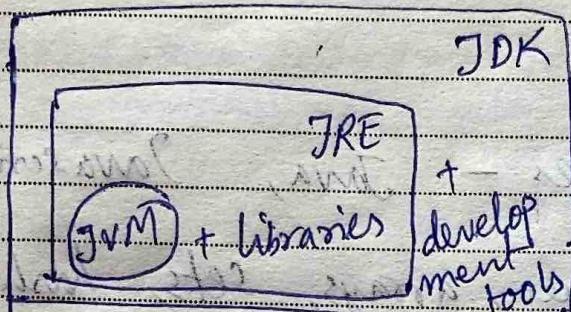
6.

Java with DSA

47/50

2022	FEBRUARY
1	2
14	15 16 17 18 19 20 21 22 23 24 25 26 27
28	
M T W T F S S M T W T F S S	

8

BasicJava

9

JDK → Java Development Kit

JRE → Java Runtime Environment

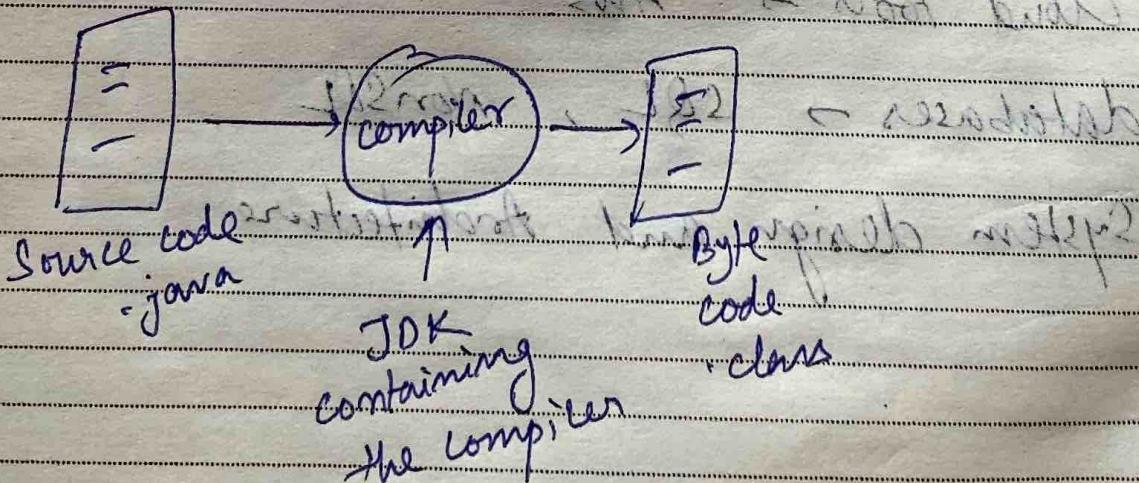
10

JVM → Java Virtual Machine

Q) How does Java code run?

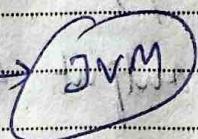
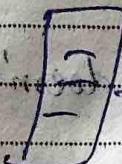
→ 2 step process - (i) compilation (ii) Execution

① compilation



JK, containing the Compiler, converts the Source code into byte code. Byte code can run in any operating system, provided there is JRE. That's why Java is also called a portable language.

⑪ Execution



native
code

By He
cable

Q) Meaning of System.out.print ("Hello World")

`System.out.print("word");` → terminates command

class → output → function printing in console

class has output → function printing

2 8) Boiler plate code - what is it?

3 → In any code editor, whatever the code we get by default at the initial stage is called boiler plate code.

Ex 1 - Package mango

Public class Hello

public static void main (String [] args)

```
{  
    System.out.print("Hello world");  
}  
}
```

2022

FEBRUARY

1	2	3	4	5	6	7	8	9	10	11	12	13
14	15	16	17	18	19	20	21	22	23	24	25	26
27	28											

M	T	W	T	F	S	S	M	T	W	T	F	S
---	---	---	---	---	---	---	---	---	---	---	---	---

17

February

8th Wk • 048-317

Thursday

Q) difference b/w different types of print command.

8 → System.out.print("hello");
 9 → prints the output but doesn't go to next line

10 System.out.println("hello");

11 → prints as well as goes to the next line

12 System.out.print("hello\n");

1 due to \n, it goes to the next line.

Q) Data types in Java?

2 data types

3 primitive

4 Non-primitive

5 byte

6 String

short

7 Array

char

8 class

boolean

9 object

int

10 long

11 interface

float

12 double

* Since Java supports primitive data type, it's not a pure object orient language.

2022

MARCH

1	2	3	4	5	6	7	8	9	10	11	12	13	
14	15	16	17	18	19	20	21	22	23	24	25	26	27
28	29	30	31										

M	T	W	T	F	S	S	M	T	W	T	F	S
---	---	---	---	---	---	---	---	---	---	---	---	---

February

8th Wk • 049-316

18

Friday

80

Java is a typed language. It means we need to define the type of the variable before defining it.

Unlike other language such as JavaScript, which is a weakly typed language, we need to define in Java.

8) Mathematical rule in Java.

→ In Java, the conventional BODMAS rule doesn't apply.

* / % has greater priority than +, -.

It goes from left to right in case of these operations.

for e.g., $a = 10, b = 5$:

Now, $a * b / a - b$;

$$= 10 * 5 / 10 - 5$$

$$= 50 / 10 - 5$$

$$= 5 - 5 = 0$$

But Parathesis gets more priority, so,

$$= (10 * 5) / (10 - 5)$$

$$= 50 / 5$$

$$= 10$$

February

8th Wk • 050-315

Saturday

1	2	3	4	5	6	7	8	9	10	11	12	13
14	15	16	17	18	19	20	21	22	23	24	25	26
27	28											
M	T	W	T	F	S	S	M	T	W	T	F	S

- Q) How to take input in Java?
- Import java.util.*; is the first step.
- class Main {
- {
- public static void main (String [] args)
- }
- System.out.println ("Arun Kumar");
- }
- we imported the java utility package and used Scanner class to take the input. 'sc' is the object of the Scanner class.
- Scanner sc = new Scanner (System.in);
- String name = sc.next();
- next() only accepts a single token. So, if it's a sentence like "My name is Tony", it will only take the first token "my" and print it.
- String name = sc.nextLine();
- accepts the whole sentence.
- int num1 = sc.nextInt();
- accepts the integer.
- float num2 = sc.nextFloat();
- accepts the floating (decimal) type number.

2022

MARCH

1	2	3	4	5	6	7	8	9	10	11	12	13
14	15	16	17	18	19	20	21	22	23	24	25	26
27	28	29	30	31								

MTWTFSSMTWTFSS

February

8th Wk • 051-314

20

Sunday

52

3) what are the conditional statements in Java?

8 → if, else
else if9
10 switch

11 break

12 Q) what is stack memory and heap memory?

→ The memory that is allocated in a pre-defined fashion is the stack memory. All the methods & variables will be stored in the stack area.

→ whereas, the memory created during the runtime for objects, arrays, etc. will be the heap memory.

e.g. class Fruits

{ public void calculation (int a)

{ return a * a;

}

public static void main (String [] args)

{

int b [] = new int [10],

calculation (b []);

}

→ calculation (), main (), int b [] will be stored in stack, whereas b [] will be stored in the heap memory.

21

February

9th Wk • 052-313

Monday

2022

53
FEBRUARY

1	2	3	4	5	6	7	8	9	10	11	12	13
14	15	16	17	18	19	20	21	22	23	24	25	26
27	28											

MTWTFSSMWTFS

Q) Pointers are used in C/C++ why does Java not use Pointers?

8

- i) Points make code more complex & ambiguous.
- ii) Security can be compromised as users can access memory allocation via pointers.
- iii) Pointers make the process of garbage collection slow and erroneous.
- iv) Java makes use of references, as these cannot be manipulated.

12

Q) global variables and local variables.

- i) global variables are variables that are defined outside of methods and can be accessed by all methods of that class.
If any change is made in any method, then only that instance of the variable is changed.

6

- ii) local variables are defined with the scope of a specific method. It can't be accessed by other method, unless defined otherwise.



Q) what are default values assigned to variables and objects/instances in Java?

- No default value for variables. we need to initialize any variables, otherwise may throw a compile time error.

MARCH											
1	2	3	4	5	6	7	8	9	10	11	12
13	14	15	16	17	18	19	20	21	22	23	24
25	26	27	28	29	30	31					
M	T	W	T	F	S	S	M	T	W	T	F

2022

February
9th Wk • 053-312

54

22
Tuesday

For any instances, default constructor may initialize the default value, depending on the type of the instance.

for reference type → null

for numeric type → 0

for boolean type → false.

Q) Difference b/w encapsulation and abstraction.

→ Encapsulation is data hiding at the implementation level.

Abstraction is data hiding at the design level.

The difference lies in the intent of the two concepts.

Abstract class

class Employee

 public int empID;
 public int salary;

 public void saveEmployee (int eid, int sal)
 {
 this.empID = eid;
 this.salary = sal;
 }

 public void printEmployee ()

 {
 System.out.println ("id" + empID + "Salary is" + salary);
 }

→ Here outside world can access saveEmployee methods but pointEmployee wouldn't know how variables are stored or printed.

1	2	3	4	5	6	7	8	9	10	11	12	13
14	15	16	17	18	19	20	21	22	23	24	25	26
27	28											
		M	T	W	T	F	S	S	M	T	W	T

23

February

9th Wk • 054-311

Wednesday

Abstraction is done by creating a class, methods, and variables of the class.

8

e.g) Encapsulation

9

class Employee

10

{

 private int empID;

11

 private long salary;

12

 public void SaveEmp (int eid, long sal)

{

1

 this.empID = eid;

2

 this.salary = sal;

}

3

 public void PrintEmp()

4

{

5

 System.out.println ("id is " + empID + " salary is " + salary);

6

}

→ Encapsulation is done by using ~~access~~ access modifiers, like `public`, `private`, `protected`, `internal`.



Hiding the data info from the outside world.

1	2	3	4	5	6	7	8	9	10	11	12	13
14	15	16	17	18	19	20	21	22	23	24	25	26
28	29	30	31									

M T W T F S S M T W T F S S

Q) JIT Compiler.

→ Just In-Time Compiler is contained inside the JVM.

✓ It basically converts the similar functionalities of the byte code into native code at the same time.

✓ Once that is done, JVM can run / execute the native code directly without the need to interpret the byte code again.

✓ That's JIT compiler increases the compile time and helps in faster execution.

Q) Difference b/w "==" operator and .equals() method.

→ "==" operator checks the memory allocation and not the actual values.

For e.g., Str1 = "mango";

Str2 = "mango";

S. o. println(Str1 == Str2); → returns true as both the strings are located in the constant pool.

Ans,

Str1 = "mango";

String Str2 = new String("mango");

S. o. println(Str1 == Str2); → returns false as

Str2 is stored in heap memory, whereas Str1 is stored in stack memory.

25

February

9th Wk • 056-309

Friday

2022	57
FEBRUARY	
1	2
3	4
5	6
7	8
9	10
11	12
13	14
15	16
17	18
19	20
21	22
23	24
25	26
27	28
M	T
W	T
F	S
S	M
T	W
F	S

Q) However, ideally it should be true because the values are same.

S. o. p. n (str1.equals (str2)); → true, bcz .equals() actually checks the values of both the strings.

Q) can the 'main' method be overloaded?

→ Yes, the main method can be overloaded as many time as we want.

But, JVM has a predefined calling method that JVM will only call the main method that has the signature like -

public static void main (String [] args)

Q) A Single try block and multiple catch blocks can co-exist in a Java program. Explain.

→ Yes, multiple catch blocks can exist with a single try block but only one catch block fulfilling the condition is executed.

try {

int a = 20, b = 0;

~~soo~~ int c = a/b;

}

catch (ArrayIndexOutOfBoundsException)

{ s. o. p. n (" white");

}

2022

MARCH	1	2	3	4	5	6	7	8	9	10	11	12	13	
	14	15	16	17	18	19	20	21	22	23	24	25	26	27
	28	29	30	31										

February
9th Wk • 057-308

26
Saturday

catch (~~Arithmetic~~ Arithmetic Exception)

{ } S-o-ph ("Black");

So, here the 2nd catch will be executed and "black" will be printed, because it is caught by division by zero, which is an arithmetic exception.

Q) Explain the use of 'final' Keyword in variable, method and class.

→ i) For variable, if final is used, value of it cannot be modified.

If any previous value not assigned, then only construction can assign it.

ii) For method, if it's declared as final, it can't be overridden by its children's classes.

A constructor cannot be marked 'final' because as when a class is inherited, constructors are not inherited. If a constructor is marked 'final', Java throws compile time error - "Modifier final not allowed".

iii) For class, if it's marked final, ~~no other~~ no other class can inherit it. But it can extend other classes. So,

final class A { }

class B extends A X

final class A { }

class A extends B ✓

27

February

9th Wk • 058-307

Sunday

2022 59
FEBRUARY

1	2	3	4	5	6	7	8	9	10	11	12	13
14	15	16	17	18	19	20	21	22	23	24	25	26
27	28											

M T W T F S S M T W T F S

Q) Difference b/w final, finally, finalize

8 ✓ final → can be marked with variable, methods or class.

9 ✓ finally → can be executed irrespective of try & catch blocks.

10 11 E.g., try {

12 } catch {

1 finally {

2 so, pln ("great");

3 So, irrespective of executing the try and catch block, the finally block() is going to be executed.

4 ✓ finalize → it is used for garbage collection.
It is called so that the clean-up activity is implemented.

5 E.g., Public static void main (String [] args) {

6 String exp = new String ("interview");
exp = null;

System.gc(); // garbage collected called

Public void finalize () {

} // finalize called for cleanup.

2022

MARCH

1	2	3	4	5	6	7	8	9	10	11	12	13	
14	15	16	17	18	19	20	21	22	23	24	25	26	27
28	29	30	31										

M	T	W	T	F	S	S	M	T	W	T	F	S
---	---	---	---	---	---	---	---	---	---	---	---	---

February

10th Wk • 059-306

28

Monday

Q) why is the main method in Java always static?

8 → Main method belongs to the class and not to any specific object if the main method is not static then any object can call it and that is not acceptable.

9 JVM calls the main method from the class name itself, not by creating any explicit object.

10 Q) can static methods be overloaded or overridden?

11 → overloading → yes, static methods can be overloaded with same name but different signatures.

12 Overriding → no, because even if child class overrides the same static method of the parent class, run-time polymorphism can't take place.

13 Q) difference between static methods, static variables and static classes in Java.

14 → Static methods → methods that belong to the class of the java program and not to any object.

15 for e.g. → Math. max(), Math. min(). max(), min(), Pow(), etc. are static methods called directly from Math class.

16 Static class → A class can't be static, unless if it's the inner class. And if it's inner class, it basically acts as the static method only.



Q) Main objective of garbage collection?

- 8 ✓ → to free up space in the memory by deleting unnecessary and unreachable objects.
- 9 This ensures efficient memory usage.
- 10 ✓ However, it provides no guarantee that there would be sufficient memory for the program execution.
- 11 ✓ Heap memory is cleaned up in the garbage collection process, not stack memory.

Q) What are shallow copy and deep copy in Java?

- 3 • Shallow copy → creates a new reference and points to the same object.
- 4 • Deep copy → creates a new object and copy the old object value to new object.

6 E.g., class Rectangle

```
{ int length = 3;
```

```
Rectangle obj1 = new Rectangle();
```

7 Rectangle obj2 = obj1; → shallow copy

```
Rectangle obj3 = new Rectangle();
```

obj3.length = obj1.length(); → deep copy.

62

MAR '22

Action Plan

01 Tue	02 Wed	03 Thu	04 Fri
Shallow copy	obj 1	length = 3	
05 Sat	06 Sun	07 Mon	08 Tue
	obj 2		
09 Wed	10 Thu	11 Fri	12 Sat
Deep copy	obj 3	15 Tue	length = 3
13 Sun			16 Wed
17 Thu	18 Fri	19 Sat	20 Sun
21 Mon	22 Tue	23 Wed	24 Thu
25 Fri	26 Sat	27 Sun	28 Mon
29 Tue	30 Wed	31 Thu	

MAR