DAY 2

Section 1: Java Data Types

1. What are the different primitive data types available in Java?

There are 8 primitive datatypes in java

Byte, short, int, long, float, double, char, Boolean

2. Explain the difference between primitive and non-primitive data types in Java.

Primitive : they are stored in the stack memory, they holds actual data values

Eg: int, char, boolean

Non Primitive : They are stored in the heap memory, they hold references to object

Eg: String, Array

3. Write a Java program that demonstrates the use of all primitive data types.

```
public class PrimitiveTypes {
   public static void main(String[] args) {
      byte b = 100;
      short s = 30000;
      int i = 100000;
      long l = 10000000000L;
      float f =8.67f;
      double d =4.6785632148;
      char c = 'A';
      boolean flag = true;
      System.out.println("byte: " + b);
      System.out.println("short: " + s);
      System.out.println("int: " + i);
      System.out.println("long: " + l);
      System.out.println("float: " + f);
      System.out.println("double: " + d);
      System.out.println("char: " + c);
      System.out.println("boolean: " + flag);
   }
}
```

4. What is type casting? Provide an example of implicit and explicit casting in Java.

It is the process of converting one datatype to another, it is of two types

Implicit: converting smaller to larger type, it is also called widening

Int num =10; double val = num;

Explicit: converting larger type to smaller type

Double d = 2.34; int I = (int) d;

5. What is the default value of each primitive data type in Java?

byte	0
short	0
int	0
long	0L
float	0.0f
double	0.0d
char	'\u0000'
boolean	false

Section 2: Java Control Statements

1. What are control statements in Java? List the types with examples.

Control statement are used to change the flow of execution

Selection statement: if, if-else, switch

Iteration: for, while, do-while

Jump statement : break, continue, return

2. Write a Java program to demonstrate the use of if-else and switch-case statements.

```
public class if_else_switch_demo {
        public static void main(String[] args) {
                // TODO Auto-generated method stub
                int num = 5;
    if (num > 0) {
       System.out.println("Positive number");
    } else {
```

```
            System.out.println("Non-positive number");
        }
        switch (num) {
            case 1: System.out.println("One"); break;
            case 5: System.out.println("Five"); break;
            default: System.out.println("Other");
        }
            }
    }
```

3. What is the difference between break and continue statements?

Break : it will exit the loop entirely, it is used to stop execution

Continue: it skips the current execution, and continues to next

4. Write a Java program to print even numbers between 1 to 50 using a for loop.

```
public class EvenNumbers {
   public static void main(String[] args) {
      for (int i = 1; i <= 50; i++) {
         if (i % 2 == 0) {
            System.out.print(i + " ");
         }}
    }
}
```

5. Explain the differences between while and do-while loops with examples.

while : checks condition first, entry controlled, only executed if the condition is statisfied

do-while : executes at least once before checking, exit controlled, executes atleast once, even if condition satisfied or not.

Section 3: Java Keywords and Operators

1. What are keywords in Java? List 10 commonly used keywords.

Keywords are some reserved words in java, which has its own meaning , we cannot use those words for any other use like variable name

public, class, static, final, if, else, switch, break, continue, return

2. Explain the purpose of the following keywords: static, final, this, super.

static: Belongs to the class, not object.

final: Makes a variable constant, a method unoverridable, or a class unextendable.

this: Refers to current object.

super: Refers to parent class methods or constructors.

3. What are the types of operators in Java?

Arithmetic: + - * / %

Relational: == != > < >= <=

Logical: && || !

Bitwise: & |

Assignment: = += -= *= /= %=

Unary: ++ --

Ternary: ?:

4. Write a Java program demonstrating the use of arithmetic, relational, and logical operators.

```
public class OperatorDemo {
    public static void main(String[] args) {
        int a = 5, b = 10;
        System.out.println(a + b);
        System.out.println(a > b);
        System.out.println((a < b) && (b > 0));
    }
}
```

5. What is operator precedence? How does it affect the outcome of expressions?

It is the priority of the Operator, Operator precedence decides which operation happens first.

int result = 10 + 2 * 5 = 20 ; in this multiplication happens first, and then the addition results in 20

but in case if addition happens first then after multiplication we will get 70 which is wrong

Additional Questions

Java Data Types

6. What is the size and range of each primitive data type in Java?

Type	Size	Range
byte	8 bits	-128 to 127
short	16 bits	-32,768 to 32,767
int	32 bits	-2,147,483,648 to 2,147,483,647
long	64 bits	About ±9 quintillion
float	32 bits	~6–7 decimal digits precision
double	64 bits	~15 decimal digits precision
char	16 bits	Single Unicode character
boolean	1 bit	true or false

7. How does Java handle overflow and underflow with numeric types?

Java will wrap around number if they go beyond their limit

byte b = 127; // max value

b++; // goes to -128 (wraps around)

System.out.println(b);

8. Write a program to convert a double value to an int without data loss.

```
public class DoubleToInt{
   public static void main(String[] args) {
      double d = 45.78;
      int i = (int) Math.round(d);
      System.out.println(i);
   }
}
```

9. What is the difference between char and String in Java?

Char : one single character like ‘A’

String : a sequence of character like “Raniya”

10. Explain wrapper classes and their use in Java.

Wrapper classes are object versions of Primitive types, so we can use primitives in Collections like ArrayList<Integer>, it also provides various useful methods

Eg: int -> Integer , double -> Double

Java Control Statements

6. Write a Java program using nested if statements.

```
public class NestedIfDemo {
   public static void main(String[] args) {
      int age = 20;
      boolean hasID = true;
      if (age >= 18) {
         if (hasID) {
            System.out.println("You can enter");
         } else {
            System.out.println("ID required");
         }
      } else {
         System.out.println("You’re too young");
      }
   }
}
```

7. Write a Java program to display the multiplication table of a number using a loop.

```
public class Table {
   public static void main(String[] args) {
      int num = 5;
      for (int i = 1; i <= 10; i++) {
         System.out.println(num + " x " + i + " = " + (num * i));
      }
   }
}
```

8. How do you exit from nested loops in Java?

By using break statements

9. Compare and contrast for, while, and do-while loops.

For: used to repeat a certain operation a fixed number of times, it will check the condition before very iteration, it will only run when the condition is satisfied

While: it is also used to repeat certain statement based on some conditions, it will also check before every iteration, runs only when the condition is satisfied

Do-while: it is also used to repeat certain statement based on some conditions, conditions will be checked after the first iteration , it will execute atleast one, even if the condition is satisfied.

10. Write a program that uses a switch-case to simulate a basic calculator

```
import java.util.Scanner;

public class Calculator {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter first number: ");
        double a = sc.nextDouble();
        System.out.print("Enter second number: ");
        double b = sc.nextDouble();
        System.out.print("Choose operation (+ - * /): ");
        char op = sc.next().charAt(0);
        switch (op) {
            case '+': System.out.println("Result: " + (a + b)); break;
            case '-': System.out.println("Result: " + (a - b)); break;
            case '*': System.out.println("Result: " + (a * b)); break;
            case '/': System.out.println("Result: " + (a / b)); break;
            default: System.out.println("Invalid operation");
        }
    }
}
```

Java Keywords and Operators

6. What is the use of the `instanceof` keyword in Java?

Checks if an object belongs to a certain class.

```
String s = "Hello";
System.out.println(s instanceof String);
```

7. Explain the difference between `==` and `.equals()` in Java.

== : compares **memory address** (are they the same object?)

.equals() : compares **values** (do they look the same?)

```
String a = new String("Hi");
String b = new String("Hi");
System.out.println(a == b);
System.out.println(a.equals(b))
```

8. Write a program using the ternary operator.

```
public class TernaryOpr {
	public static void main(String[] args){
		int age = 20;
		String result = (age >= 18) ? "Adult" : "Minor";
		System.out.println(result);
	}
}
```

9. What is the use of `this` and `super` in method overriding?

This: calls the current class's stuff

Super: calls parent class's stuff

```
class Parent {
   void greet() { System.out.println("Hello from Parent"); }
}
class Child extends Parent {
   void greet() {
      super.greet();
      System.out.println("Hello from Child");
```

```
    }
}
```

10. Explain bitwise operators with examples.

They work on binary level

```
int a = 5;
int b = 3;
System.out.println(a & b);
System.out.println(a | b);
System.out.println(a ^ b);
```