Section 1: Java Data Types

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

Ans: Java has 8 primitive datatypes those are used to data like numbers, characters and true/false values.

1. int
2. byte
3. short
4. long
5. float
6. double
7. char
8. Boolean

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

- **Primitive data types:**It can store a single value directly in memory. They are predefined by Java and don't have methods.
- **Non-primitive data types** (reference types):It can store a memory address that points to an object. They are created by the programmer.

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

Ans:

Program:

```
public class Test {

    public static void main(String[] args) {

        byte b = 100;

        short s = 2000;

        int i = 100000;

        long l = 123456789L;

        float f = 5.75f;

        double d = 19.99;

        char c = 'A';

        boolean bool = true;

        System.out.println(b);

        System.out.println(s);

        System.out.println(i);
```

```
        System.out.println(l);
        System.out.println(f);
        System.out.println(d);
        System.out.println(c);
        System.out.println(bool);
    }
}
```

Output :

100

2000

100000

123456789

5.75

19.99

A

true

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

Ans: Typecasting means converting one datatype to another datatype. There are two types

1. Widening : Converting small datatype to large datatype which done by automatically.

Ex: int -> long -> float

2. Narrowing : Converting larger datatype to smaller datatype which done by manually.

Ex: float -> long -> int

```
public class Casting {
    public static void main(String[] args) {
        int num = 20;
        System.out.println("value of num : " + num);
        double d = num;
        System.out.println("value of d : " + d);
```

```
        int a  = (int) d;
        System.out.println("value of a : " + a);
    }
}
```

Output :

value of num : 20

value of d :20.0

value of a : 20

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

Ans: When a primitive type is declared as an instance variable, Java assigns a default value if not initialized.

1. byte – 0
2. short – 0
3. int – 0
4. long – 0L
5. float – 0.0f
6. double – 0.0d
7. char – \u
8. boolean - false

Section 2 : Java Control Statements

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

Ans: Control statements are used to control the flow of execution of the program.

1. Conditional Statements : if, if-else, if-else-if(ladder), nested if, switch
2. Looping Statements: while, do-while, for
3. Jumping statements : break, continue, return

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

```
public class ControlStatementsDemo {
   public static void main(String[] args) {
        int num = 18;
        if (num % 2 == 0) {
            System.out.println(num + " is even.");
        } else {
            System.out.println(num + " is odd.");
```

```
        }
        int val = 2;
        switch(val) {
                case 1: System.out.println("Tea");
                break;
                case 2: System.out.println("Coffee");
                break;
                case 3: System.out.println("Juice");
                break;
                default: System.out.println("Invalid Input ");
        }
    }
}
```

Output :

18 is even

Coffee

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

- Break exits the entire loop or switch block.
- continue skips the current iteration and proceeds to the next one.

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 + " ");
            }
        }
    }
}
```

Output: 2 4 6 8 10 12 14 16 18 20 22 24 26 28 30 32 34 36 38 40 42 44 46 48 50

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

- Ans: A **while loop** checks the condition at the beginning and may not execute if the condition is initially false.
- A **do-while loop** executes the body at least once, as the condition is checked at the end.

Program:

```
public class Test {
public static void main(String[] args) {
        int a = 1;
        while (a < 5) {
                System.out.println(a+ " ");
                a++;
         }
        int b = 1;
        do {
                System.out.print(b+ " ");
                b++;
                } while (b < 5);
        }
}
```

Output: 1 2 3 4 1 2 3 4

Section 3 : Java Keywords and Operators

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

Ans: Java keywords are reserved words that have predefined meanings in Java.

ex : if, else, for, while, try, throw, throws, catch, int, char, float, static, public, private,

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

Ans:

Static : 1.It is belongs to class, not to instance

2. shared across all objects of class

3. useds for variable, methods

Final : 1.used to create constants

2. used for variables, methods

This : 1. Refers to the current object

2. used to avoid name conflicts when there is same name.

Super: 1. Refers to parent class

2. used to access parent class variables and methods

3. What are the types of operators in Java?

Ans: Operators are symbols used to perform operations on variables and values.

1. Arithmetic Operators
2. Relational Operators
3. Logical Operators
4. Assignment Operators
5. Unary Operators
6. Bitwise Operators
7. Shift Operators
8. instanceOf Operators

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

Ans :

Program :

```
public class Test {
public static void main(String[] args) {
    int a = 10, b = 20;
    // Arithmetic
    System.out.println(a + b);
    // Relational
    System.out.println(a > b);
    // Logical
    System.out.println(a > 5 && b > 15);
  }
}
```

Output : 30

false

true

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

Ans : Operator precedence is priority of operators in expression. It decides which operation should execure first when multiple operators are in same expression

ex :

```
public class Test {
public static void main(String[] args) {
    int result = 10 + 5 * 2;
    System.out.println("result :  " +result);
  }
}
```

Output : 20