CLASS -12 (2025-26)

JAVA RIVISION TOUR

CHAPTER 4

Assignments:-

1.

Statement 1:

```
number = number + number / number;
```

- `number / number` evaluates to `1` (since any non-zero number divided by itself is 1).
- So, the result is:
  `number = number + 1;`

Statement 2:

```
number += number / number;
```

- This is a shorthand for:
  `number = number + (number / number);`
- So again, this becomes:
  `number = number + 1;`

Statement 3:

```
number = number + 7 / 7;
```

- `7 / 7` equals `1`, so this becomes:
  `number = number + 1;`

✓ Conclusion:

Yes, all three statements achieve the same thing:
They **increment `number` by 1**, as long as `number > 0`.

2. Write two more statements that achieve the same as `number = number + 7 / 7;`

```
number += 1;
number = number + 1;
```

Both add 1 to `number`.

---

3. How do computers represent data?

Computers represent data using **binary (0s and 1s)**. Each type of data (characters, numbers, images, etc.) is encoded in a specific binary format.

---

4. What are data types? What do they do?

Data types define the kind of data a variable can hold (e.g., `int`, `float`, `char`, `boolean`).
They help the compiler allocate memory and perform type-checking.

---

5. Difference between `short` and `int`:

Feature	`short`	`int`
Size	16-bit	32-bit
Range	-32,768 to 32,767	-2,147,483,648 to 2,147,483,647
Use	Saves memory in large arrays	Default for integers

---

6. What is an identifier? Java Identifier Rules:

An **identifier** is the name of a variable, method, class, etc.
Rules:

- Must start with a letter, `$`, or _
- Cannot start with a digit

- No spaces or special symbols except `$` and _
- Cannot be a Java keyword
- Case-sensitive

---

7. `a = 5 / 3` Result if:

- **(i)** **`float a`** → `a = 1.0` (integer division first, result is 1, then converted to float)
- **(ii)** **`int a`** → `a = 1` (standard integer division)

---

8. `j = --k + 2 * k + (l = k, l++)` if `k = 20`

Let's solve step-by-step:

- `--k` → `k = 19`
- `2 * k` → `2 * 19 = 38`
- `(l = k, l++)` → `l = 19; use 19, then l becomes 20`

So,
`j = 19 + 38 + 19 = 76`

---

9. If `j = 5` initially:

(1) `(5 * ++j) % 6`

- `++j = 6`, so: `5 * 6 = 30 % 6 = 0`

(2) `(5 * j++) % 6`

- `j = 5`, then post-increment, so: `5 * 5 = 25 % 6 = 1`

---

10. `age > 65 ? 350 : 100`

(i) `age = 25` → Result: **100**
(ii) `age = 65` → Result: **100**
(iii) `age = 85` → Result: **350**

---

11. `ans - val < 500 ? 150 : 50`

(i) `ans = 700, val = 300` → `700 - 300 = 400 < 500` → **150**
(ii) `ans = 800, val = 700` → `800 - 700 = 100 < 500` → **150**

---

12. Which of the following are correct?

(a) `int a = 16; a >> 2 = 4` → **Correct**
(b) `int b = -8; b >> 1 = -4` → **Correct**
(c) `int a = 16; a >>> 2 = 4` → **Correct**
Answer: (d) All of the above

---

13. Valid boolean expressions:

- (a) `b = x1 * 2 == x2;` → `100 * 2 == 200` → **true**
- (b) `b = x1 + x2 != 3 * x1;` → `100 + 200 != 300` → `300 != 300` → **false**
- (c) Invalid syntax: `**` is not a valid operator.
- (d) Invalid syntax: misuse of assignment in condition.

Answer: (a) is correct

---

14. x evaluated to 8:

Let's solve each:
(a) `int x = 32; x = x >> 33;` → Invalid shift (shifts by 33 % 32 = 1) → x = 16
(b) `int x = 33; x = x >> 2;` → 33 >> 2 = 8
(c) `int x = 35; x = x >> 2;` → 35 >> 2 = 8
(d) `int x = 16; x = x >> 1;` → 16 >> 1 = 8
Answer: (b), (c), (d)

15. Write Java code

(i) Append value:

```
int x = 3;
System.out.println("x = " + x);
```

(ii) Print 3x3 matrix:

```
int a = 1, b = 2, c = 3;
int d = 4, e = 5, f = 6;
int g = 7, h = 8, i = 9;

System.out.println(a + " " + b + " " + c);
System.out.println(d + " " + e + " " + f);
System.out.println(g + " " + h + " " + i);
```

16. Variable Declarations

```
int miles = 0;
double flowRate = 50.56;
```

17. Assignment for Interest Calculation

```
interest = balance * rate;
```

(Assuming all variables are declared as `double`)

18. Input a Value from Keyboard

```
import java.util.Scanner;

Scanner scanner = new Scanner(System.in);
double amount = scanner.nextDouble();
```

19. Predicting and Testing Expressions

```
public class TestIncrements {
    public static void main(String[] args) {
        int i, j, k;

        i = 1;
        j = 3;
        k = j-- + ++i; // 3 + 2 = 5, j becomes 2
        System.out.println("1. i=" + i + " j=" + j + " k=" + k);

        i = 1;
        j = 3;
        k = ++i + j--; // 2 + 3 = 5, j becomes 2
        System.out.println("2. i=" + i + " j=" + j + " k=" + k);

        j = 3;
        k = j-- + ++j; // 3 + 3 = 6, j becomes 3
        System.out.println("3. j=" + j + " k=" + k);

        j = 3;
        k = --j + j++; // 2 + 2 = 4, j becomes 3
        System.out.println("4. j=" + j + " k=" + k);
    }
}
```

20. Predict Output and Test

```
public class LoopTest {
    public static void main(String[] args) {
        int j = 0;
        int i = 1;
        for (; i < 2; i++) j++;
        System.out.println("i=" + i + " j=" + j); // i=2, j=1

        i = 0;
        j = 5;
        do {
            j--;
        } while ((i++) < 5);
        System.out.println("i=" + i + " j=" + j); // i=6, j=0
```

```
        System.out.println("1=" + 1 + " j=" + 1); // 1=1 j=1

        j = 5;
        i = 0;
        while ((i++) < 5) {
            j--;
        }
        System.out.println("i=" + i + " j=" + j); // i=6, j=0
    }
}
```

21. Code Output Prediction

Code:

```
class Test {
    public static void main(String[] args) {
        int x = 20;
        String sup = (x < 15) ? "small" : (x < 22) ? "tiny" : "huge";
        System.out.println(sup);
    }
}
```

Answer: B. tiny

Explanation:

- `x = 20`, not less than 15, but less than 22 → chooses `"tiny"`

22. Celsius to Fahrenheit Converter

```
public class TempConverter {
    public static void main(String[] args) {
        double F = Double.parseDouble(args[0]);
        double C = (5.0 / 9.0) * (F - 32);
        System.out.println("Celsius temperature = " + C);
    }
}
```

To run in **BlueJ**, right-click the class → select `void main(String[] args)` → provide input like `{"98.6"}`

23. Count Words in a Text File

```
import java.io.File;
import java.io.FileNotFoundException;
import java.util.Scanner;

public class WordCounter {
    public static void main(String[] args) {
        try {
            File file = new File("input.txt"); // Ensure file exists
            Scanner sc = new Scanner(file);
            int wordCount = 0;

            while (sc.hasNext()) {
                sc.next();
                wordCount++;
            }

            sc.close();
            System.out.println("Total words: " + wordCount);
        } catch (FileNotFoundException e) {
            System.out.println("File not found.");
        }
    }
}
```

Place a text file named `input.txt` in the project folder to test.

****