A. TICK (✓) THE CORRECT ANSWER

1. Which of the following is the correct syntax of Java if statement?

   Answer: **a. if(condition) //statement;**

   Explanation:

   For a single statement, Java allows `if(condition) statement;` without braces.

   Option (b) has a semicolon immediately after `if(condition);` so the if has an empty body and the block after it runs always.

   Option (c) is syntactically wrong because the braces with `else` are not correct.

   Hence (a) is the proper syntax.

2. Output of:

```
if(true)
    System.out.println("Yes");
else
    System.out.println("No");
```

   Answer: **b. Yes**

   Explanation: The condition is `true`, so the `if` part executes and prints `Yes`. The `else` part is skipped.

3. Output of:

```
double a = 15.4;
if (a == 15)
{
    System.out.println("India ");
    System.out.println("New Delhi");
}
else
    System.out.println("All");
```

   Answer: **d. None of these**

   Explanation: `a` is 15.4, so `a == 15` is false. The `else` executes and prints `All`. This is not in options (a), (b) or (c), so the correct option is (d).

4. Output of:

```
float marks = 89.5f;
if(marks > 89.5)
    System.out.println("Grade A");
else
    System.out.println("Grade B");
```

   Answer: **b. Grade B**

   Explanation: `marks` is 89.5 so the test `marks > 89.5` is false (it is equal, not greater). Therefore the `else` part runs and prints `Grade B`.

5. Output of:

```
int a = 2;
switch(a)
{
    case 1: System.out.print("Tiger");
    case 2: System.out.print("Deer");
    default: System.out.println("Lion");
}
```

   Answer: **c. DeerLion**

   Explanation: `a` is 2, so control goes to `case 2`. There is no `break`, so it prints `"Deer"` and then continues to `default`, printing `"Lion"`. Final output is `DeerLion`.

6. If _____ is not present in a switch case program, then fall-through occurs.

   Answer: **c. break**

   Explanation: Without `break`, control falls through to the next case(s).

7. In Java, instead of _____, Ternary operator may be used.

   Answer: **b. if...else statement**

Explanation: The conditional or ternary operator `?:` is a compact form of a simple `if...else`.

8. The default case in the switch statement is executed only if _____.

   Answer: **b. None of the cases are matched**

   Explanation: `default` runs when no other `case` value matches the switch expression.

9. Consider:

```
switch(true)
{
    case 1:
        System.out.println("Orange");
        break;
    case 2:
        System.out.println("Education");
        break;
    default:
        System.out.println("Wrong choice");
}
```

   Answer: **d. Incompatible types error**

   Explanation: The switch expression is `true` (boolean) but the `case` labels are integers. In Java this is a compile-time type-mismatch error.

10. Code:

```
if(Name == "Orange Education")
    s = "You are at right place.";
else
    s = "Try again";
```

    Equivalent statement:

    Answer:

    a. s = (Name == "Orange Education") ? "You are at right place." : "Try again";

    Explanation: This uses the ternary operator with the same condition and the same two possible strings. Other options are syntactically wrong.

11. Which of the following is correct about switch and if statements?

    Answer: **c. Both the statements are correct**

    Explanation: A `switch` can be written inside an `if`, and an `if` can also be written inside a `case` of a `switch`. Both nestings are allowed.

B. FILL IN THE BLANKS

1. Java Ternary operator is also called **conditional operator**.
2. The absence of break in **switch** statement is known as fall-through.
3. One example of **selection** statement is a switch case statement.
4. The `if(a == 0)` will return **false** when `a` is a non-zero value.
5. The `if...else` statement always returns **boolean** data type (its condition is a boolean expression).

C. SHORT ANSWER TYPE QUESTIONS

1. What is the difference between if and if...else statements?

   Answer:

   `if` statement executes a block only when the condition is true; nothing happens when it is false. `if...else` statement provides two alternative blocks: one block executes when the condition is true and another block executes when the condition is false.

2. What are compound statements?

   Answer:

   A compound statement is a group of one or more statements enclosed within a pair of braces `{}` and treated as a single statement, for example:

```
    {
        x = x + 1;
        y = y - 1;
    }
```

3. What is a fall-through situation?

   Answer:

   In a `switch` statement, if a `case` does not end with a `break` and the next case statements (including `default`) start executing automatically, this is called a fall-through situation.

4. What is the use of the break statement in the switch statement?

   Answer:

   `break` terminates the current `case` and transfers control out of the entire `switch` block. It prevents fall-through to the next case.

5. Program: input a number and check whether it is divisible by 7 and 8.

```
import java.util.Scanner;

class Q5_Divisible7and8
{
    public static void main(String[] args)
    {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter a number: ");
        int n = sc.nextInt();

        if(n % 7 == 0 && n % 8 == 0)
            System.out.println("Divisible by both 7 and 8");
        else
            System.out.println("Not divisible by both 7 and 8");
    }
}
```

   Explanation: We use the modulus operator `%`. If remainder with both 7 and 8 is zero, the number is divisible by both.

6. Program: input a year (four digits) and print whether it is a leap year or not.
   Condition: A leap year is (multiple of 400) OR (multiple of 4 and not multiple of 100).

```
import java.util.Scanner;

class Q6_LeapYear
{
    public static void main(String[] args)
    {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter a year (4 digits): ");
        int year = sc.nextInt();

        if( (year % 400 == 0) || (year % 4 == 0 && year % 100 != 0) )
            System.out.println("Leap year");
        else
            System.out.println("Not a leap year");
    }
}
```

7. Program: input three numbers and print the largest and the smallest.

```
import java.util.Scanner;

class Q7_LargestSmallest
{
    public static void main(String[] args)
    {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter three numbers: ");
        int a = sc.nextInt();
```

```
            int b = sc.nextInt();
            int c = sc.nextInt();

            int largest = a;
            if(b > largest)
                largest = b;
            if(c > largest)
                largest = c;

            int smallest = a;
            if(b < smallest)
                smallest = b;
            if(c < smallest)
                smallest = c;

            System.out.println("Largest = " + largest);
            System.out.println("Smallest = " + smallest);
        }
    }
```

Explanation: We assume `a` as largest and smallest first, then compare with `b` and `c`.

8. Shopping mall discount program.

Discount:

- Up to 2000 : 2.5%
- 2001 to 4000 : 4%
- 4001 to 7000 : 7%
- More than 7000 : 10%

```
import java.util.Scanner;

class Q8_MallDiscount
{
    public static void main(String[] args)
    {
        Scanner sc = new Scanner(System.in);

        System.out.print("Enter customer name: ");
        String name = sc.nextLine();

        System.out.print("Enter purchase amount: ");
        double amount = sc.nextDouble();

        double rate;

        if(amount <= 2000)
            rate = 2.5;
        else if(amount <= 4000)
            rate = 4.0;
        else if(amount <= 7000)
            rate = 7.0;
        else
            rate = 10.0;

        double discount = amount * rate / 100.0;
        double netAmount = amount - discount;

        System.out.println("Customer Name: " + name);
        System.out.println("Purchase Amount: " + amount);
        System.out.println("Discount: " + discount);
        System.out.println("Amount to be paid: " + netAmount);
    }
}
```

LONG PROGRAM QUESTIONS (9 TO 15)

9. Telephone bill on the basis of number of calls:

Slab (number of calls) and rate per call:
More than 250 calls : Rs. 5 per call
150 to 249 calls : Rs. 4 per call
75 to 149 calls : Rs. 3 per call
Less than 75 calls : Rs. 2 per call
Surcharge: 2.5% on bill amount.

```
import java.util.Scanner;

class Q9_TelephoneBill
{
    public static void main(String[] args)
    {
        Scanner sc = new Scanner(System.in);

        System.out.print("Enter consumer name: ");
        String name = sc.nextLine();

        System.out.print("Enter consumer number: ");
        int cno = sc.nextInt();

        System.out.print("Enter number of calls: ");
        int calls = sc.nextInt();

        double rate;

        if(calls > 250)
            rate = 5.0;
        else if(calls >= 150)
            rate = 4.0;
        else if(calls >= 75)
            rate = 3.0;
        else
            rate = 2.0;

        double billBefore = calls * rate;
        double surcharge = billBefore * 2.5 / 100.0;
        double billWithSurcharge = billBefore + surcharge;

        System.out.println("Name of consumer: " + name);
        System.out.println("Consumer Number: " + cno);
        System.out.println("Number of calls: " + calls);
        System.out.println("Bill amount before surcharge added: " +
billBefore);
        System.out.println("Bill amount with surcharge: " +
billWithSurcharge);
    }
}
```

Explanation: Choose the rate according to calls, compute bill, add 2.5% surcharge.

10. Mode of payment and facilities:

Mode and facilities:

- Credit Card (cc) : 1.5% discount
- Debit Card (dc) : Rs. 10 cashback
- E Wallet (ew) : Rs. 20 cashback
- Cash (c) : No discount

```
import java.util.Scanner;
```

```
class Q10_ModeOfPayment
{
    public static void main(String[] args)
    {
        Scanner sc = new Scanner(System.in);

        System.out.print("Enter bill amount: ");
        double amount = sc.nextDouble();

        System.out.print("Enter mode of payment (cc/dc/ew/c): ");
        String mode = sc.next();

        double netAmount = amount;

        if(mode.equalsIgnoreCase("cc"))
        {
            double discount = amount * 1.5 / 100.0;
            netAmount = amount - discount;
            System.out.println("Facility: 1.5% discount = " + discount);
        }
        else if(mode.equalsIgnoreCase("dc"))
        {
            netAmount = amount - 10.0;
            System.out.println("Facility: Rs. 10 cashback");
        }
        else if(mode.equalsIgnoreCase("ew"))
        {
            netAmount = amount - 20.0;
            System.out.println("Facility: Rs. 20 cashback");
        }
        else if(mode.equalsIgnoreCase("c"))
        {
            System.out.println("Facility: No discount");
        }
        else
        {
            System.out.println("Invalid mode, amount unchanged");
        }

        System.out.println("Amount to be paid: " + netAmount);
    }
}
```

11. Electricity bill based on units consumed.

Rate:

- First 100 units : Rs. 2 per unit
- Next 150 units : Rs. 3.5 per unit (units 101 to 250)
- Next 400 units : Rs. 5 per unit (units 251 to 650)
- More than 650 units : Rs. 7.5 per unit (units above 650)

```
import java.util.Scanner;

class Q11_ElectricityBill
{
    public static void main(String[] args)
    {
        Scanner sc = new Scanner(System.in);

        System.out.print("Enter previous month reading: ");
        int prev = sc.nextInt();

        System.out.print("Enter current month reading: ");
```

```
            int curr = sc.nextInt();

            int units = curr - prev;
            double amount = 0.0;

            int remaining = units;

            if(remaining > 0)
            {
                int slab = Math.min(remaining, 100);
                amount += slab * 2.0;
                remaining -= slab;
            }

            if(remaining > 0)
            {
                int slab = Math.min(remaining, 150);
                amount += slab * 3.5;
                remaining -= slab;
            }

            if(remaining > 0)
            {
                int slab = Math.min(remaining, 400);
                amount += slab * 5.0;
                remaining -= slab;
            }

            if(remaining > 0)
            {
                amount += remaining * 7.5;
            }

            System.out.println("Units consumed: " + units);
            System.out.println("Total amount to be paid: " + amount);
        }
    }
```

Explanation: We deduct units slab by slab and accumulate the cost.

12. Company commission and gift for sales personnel.

Number of sales and commission:

- Up to 50 products : 5% on selling amount, gift = "A Parker pen"
- 51 to 75 products : 7.5% on selling amount, gift = "A Micro SD card"
- 76 to 100 products : 10% on selling amount, gift = "A Mobile"
- More than 100 products : 15% on selling amount, gift = "A Laptop"

```
import java.util.Scanner;

class Q12_SalesCommission
{
    public static void main(String[] args)
    {
        Scanner sc = new Scanner(System.in);

        System.out.print("Enter number of products sold: ");
        int sales = sc.nextInt();

        System.out.print("Enter selling amount: ");
        double amount = sc.nextDouble();

        double rate;
        String gift;
```

```
            if(sales <= 50)
            {
                rate = 5.0;
                gift = "A Parker pen";
            }
            else if(sales <= 75)
            {
                rate = 7.5;
                gift = "A Micro SD card";
            }
            else if(sales <= 100)
            {
                rate = 10.0;
                gift = "A Mobile";
            }
            else
            {
                rate = 15.0;
                gift = "A Laptop";
            }

            double commission = amount * rate / 100.0;

            System.out.println("Commission percentage: " + rate + "%");
            System.out.println("Commission amount: " + commission);
            System.out.println("Gift received: " + gift);
        }
    }
```

13. Electronics shop seasonal discounts.

Table (approx from question):

Purchase amount in Rs. | Discount on Laptop | Discount on Desktop PC

0 – 25000 | 0.0% | 5.0%

25001 – 57000 | 5.0% | 7.6%

57001 – 100000 | 7.5% | 10%

More than 100000 | 10.0% | 15.0%

```
import java.util.Scanner;

    class Q13_ElectronicShop
    {
        public static void main(String[] args)
        {
            Scanner sc = new Scanner(System.in);
            System.out.print("Enter customer name: ");
            String name = sc.nextLine();
            System.out.print("Enter address: ");
            String address = sc.nextLine();
            System.out.print("Enter amount of purchase: ");
            double amount = sc.nextDouble();
            System.out.print("Enter type of purchase (L for Laptop, D for
    Desktop): ");
            char type = sc.next().charAt(0);
            double rate = 0.0;
            if(type == 'L' || type == 'l')
            {
                if(amount <= 25000)
                    rate = 0.0;
                else if(amount <= 57000)
                    rate = 5.0;
                else if(amount <= 100000)
```

```
                rate = 7.5;
            else
                rate = 10.0;
        }
        else if(type == 'D' || type == 'd')
        {
            if(amount <= 25000)
                rate = 5.0;
            else if(amount <= 57000)
                rate = 7.6;
            else if(amount <= 100000)
                rate = 10.0;
            else
                rate = 15.0;
        }
        double discount = amount * rate / 100.0;
        double netAmount = amount - discount;
        System.out.println("Customer Name: " + name);
        System.out.println("Address: " + address);
        System.out.println("Amount of purchase: " + amount);
        System.out.println("Type of purchase: " + type);
        System.out.println("Discount: " + discount);
        System.out.println("Net amount to be paid: " + netAmount);
    }
}
```

14. Menu driven program – volume of sphere, cylinder and cone.

Formulas:

- Volume of sphere (vs) = 4/3 * pi * r^3
- Volume of cylinder (vc) = pi * r^2 * h
- Volume of cone (vco) = 1/3 * pi * r^2 * h

```
import java.util.Scanner;

class Q14_VolumeMenu
{
    public static void main(String[] args)
    {
        Scanner sc = new Scanner(System.in);
        int choice;
        double r, h, volume;
        final double PI = 3.14159;
        System.out.println("1. Volume of sphere");
        System.out.println("2. Volume of cylinder");
        System.out.println("3. Volume of cone");
        System.out.print("Enter your choice (1-3): ");
        choice = sc.nextInt();

        switch(choice)
        {
            case 1:
                System.out.print("Enter radius: ");
                r = sc.nextDouble();
                volume = (4.0 / 3.0) * PI * r * r * r;
                System.out.println("Volume of sphere = " + volume);
                break;
            case 2:
                System.out.print("Enter radius and height: ");
                r = sc.nextDouble();
                h = sc.nextDouble();
                volume = PI * r * r * h;
                System.out.println("Volume of cylinder = " + volume);
                break;
```

```
                case 3:
                    System.out.print("Enter radius and height: ");
                    r = sc.nextDouble();
                    h = sc.nextDouble();
                    volume = (1.0 / 3.0) * PI * r * r * h;
                    System.out.println("Volume of cone = " + volume);
                    break;
                default:
                    System.out.println("Wrong choice");
            }
        }
    }
```

15. Menu driven temperature converter (Fahrenheit and Celsius) using switch.

Formulae:

C = (F - 32) * 5 / 9

F = 1.8 * C + 32

```
import java.util.Scanner;
class Q15_TemperatureMenu
{
    public static void main(String[] args)
    {
        Scanner sc = new Scanner(System.in);

        System.out.println("1. Fahrenheit to Celsius");
        System.out.println("2. Celsius to Fahrenheit");
        System.out.print("Enter your choice (1 or 2): ");
        int choice = sc.nextInt();
        switch(choice)
        {
            case 1:
                System.out.print("Enter temperature in Fahrenheit: ");
                double F = sc.nextDouble();
                double C = (F - 32) * 5.0 / 9.0;
                System.out.println("Temperature in Celsius = " + C);
                break;
            case 2:
                System.out.print("Enter temperature in Celsius: ");
                C = sc.nextDouble();
                F = 1.8 * C + 32;
                System.out.println("Temperature in Fahrenheit = " + F);
                break;
            default:
                System.out.println("Wrong choice");
        }
    }
}
```

16. Menu driven class: check whether a number is palindrome or perfect.

```
import java.util.Scanner;

class Q16_PalindromePerfect
{
    public static void main(String[] args)
    {
        Scanner sc = new Scanner(System.in);
        System.out.println("1. Check palindrome number");
        System.out.println("2. Check perfect number");
        System.out.print("Enter your choice (1 or 2): ");
        int choice = sc.nextInt();

        System.out.print("Enter the number: ");
        int n = sc.nextInt();
```

```
            switch(choice)
            {
                case 1:
                    int temp = n, rev = 0;
                    while(temp > 0)
                    {
                        int d = temp % 10;
                        rev = rev * 10 + d;
                        temp = temp / 10;
                    }
                    if(rev == n)
                        System.out.println("Palindrome number");
                    else
                        System.out.println("Not a palindrome number");
                    break;

                case 2:
                    int sum = 0;
                    for(int i = 1; i <= n / 2; i++)
                    {
                        if(n % i == 0)
                            sum += i;
                    }
                    if(sum == n)
                        System.out.println("Perfect number");
                    else
                        System.out.println("Not a perfect number");
                    break;

                default:
                    System.out.println("Wrong choice");
            }
        }
    }
```

Explanation:

Palindrome: reverse digits and compare with original.

Perfect: find sum of all proper divisors (excluding the number itself) and compare with number.

17. Program: Check BUZZ number OR print GCD of two numbers.

BUZZ number: ends with 7 or divisible by 7.

GCD: Greatest Common Divisor of two integers.

```
import java.util.Scanner;

class Q17_BuzzOrGCD
{
    public static void main(String[] args)
    {
        Scanner sc = new Scanner(System.in);
        System.out.println("1. Check BUZZ number");
        System.out.println("2. GCD of two numbers");
        System.out.print("Enter your choice (1 or 2): ");
        int choice = sc.nextInt();
        switch(choice)
        {
            case 1:
                System.out.print("Enter a number: ");
                int n = sc.nextInt();
                if(n % 10 == 7 || n % 7 == 0)
                    System.out.println("BUZZ number");
                else
                    System.out.println("Not a BUZZ number");
                break;
```

```
                    case 2:
                        System.out.print("Enter two numbers: ");
                        int a = sc.nextInt();
                        int b = sc.nextInt();

                        int x = a;
                        int y = b;

                        // Euclidean algorithm for GCD
                        while(y != 0)
                        {
                            int rem = x % y;
                            x = y;
                            y = rem;
                        }
                        System.out.println("GCD = " + x);
                        break;
                    default:
                        System.out.println("Wrong choice");
                }
            }
        }
```

D. ASSERTION AND REASONING

Assertion (A): The if statement inside another if statement is called nested if.
Reason (R): The inner if statement will execute only if the outer if statement evaluates to true.

Correct option: **a. Both Assertion (A) and Reason (R) are true, and Reason (R) is a correct explanation of Assertion (A).**

Explanation:
Nested if means one if inside another. The inner if can be reached only when the condition of the outer if is true; therefore R explains A.

PREVIOUS YEARS' QUESTIONS – SECTION A

1. The absence of which statement leads to fall through situation in switch case statement?

   Answer: **b. break**

   Explanation: Without `break`, control passes to the next case(s).

2. Choose the odd one: `if else`, `if`, `switch case`, `while()`

   Answer: **d. while()**

   Explanation: `if`, `if else` and `switch` are selection (decision) statements; `while` is a looping statement.

3.
```
int f = 10, m = 9;
String e = (m % f == 9) ? "YES" : "NO";
System.out.print(e);
```
   Answer: **a. YES**

   Explanation: `m % f` is `9 % 10 = 9`. Condition is true, so e becomes `"YES"`.

4.
```
switch(x)
{
    case 'M':
        System.out.print("Microsoft Teams"); break;
    case 'G':
        System.out.print("Google Meet");
    default :
        System.out.print("Any software");
        break;
```

```
        case 'W':
            System.out.print("Web Ex");
            break;
}
```

When `x = 'g'`:

Answer: **b. Any software**

Explanation: Character `'g'` (small g) does not match `'M'`, `'G'` or `'W'`, so control goes directly to `default` and prints `"Any software"`.