

Class 10

Chapter – 10

User-Defined Methods

A. TICK (✓) THE CORRECT ANSWER

☐ A. Tick (✓) the correct answer

Q1.

An impure method is also called a _____.

- a. Mutator
- b. Class
- c. Object
- d. Functionally dependent method

✓ Answer: a. Mutator

💡 Explanation:

- Impure method → changes object state
- Such methods are called **Mutators**

Q2.

Overloaded method _____ written in same class.

- a. should not be
- b. cannot be
- c. should be
- d. is optionally

✓ Answer: c. should be

💡 Explanation:

- Method overloading happens **within same class**

Q3.

Polymorphism is one of the features of _____ programming language.

- a. Procedure-Oriented
- b. High-Level

- c. Object-Oriented
- d. Low-Level

✔ Answer: c. Object-Oriented

Q4.

In which of the following, any change in the formal parameter will not reflect on the actual parameter?

- a. Pass by value
- b. Pass by reference
- c. Class
- d. Object

✔ Answer: a. Pass by value

💡 Explanation:

- Java uses **pass by value**
- Changes inside method do NOT affect original variable

Q5.

In which of the following, any change in the formal parameter while the method is executing, the actual parameter also changes?

- a. Pure method
- b. Impure Method
- c. Pass by value
- d. Pass by object

✔ Answer: d. Pass by object

Q6.

What is method overloading in Java?

- a. Same name, different return types
- b. Same name, different parameters
- c. Different access modifiers
- d. Different names

✔ Answer: b

Q7.

Which of the following is true about method overloading?

- a. Not allowed
- b. Only return type differs
- c. Same type parameters required
- d. Same method name required

✓ Answer: d

Q8.

What is the benefit of using user-defined methods in Java?

- a. Reduces code duplication and increases reusability
- b. Increases complexity
- c. Decreases execution speed
- d. Makes program harder

✓ Answer: a

Q9.

Which is correct syntax of method declaration?

- a. public static void myMethod() {}
- b. static void my Method() {}
- c. void myMethod(integer x) {}
- d. void int myMethod() {}

✓ Answer: a

Q10.

Can a method call itself in Java?

- a. Yes, only static
- b. No
- c. Yes, recursion
- d. Yes, only public

✓ Answer: c

💡 Explanation:

- Method calling itself = **Recursion**

□ **B. Fill in the blanks**

Q1.

_____ occupy less space in memory and execution becomes faster.

✓ Answer: Methods

Q2.

Every method should be provided with a _____.

✓ Answer: name

Q3.

Methods already defined in Java packages are called _____ methods.

✓ Answer: predefined

Q4.

By default, the _____ statement returns a value.

✓ Answer: return

Q5.

Only _____ data can be used in pass by value.

✓ Answer: primitive

● C. Assertion and Reasoning Based Questions

Options:

- a. Both A and R true, R explains A
- b. Both true, R not explanation
- c. A true, R false
- d. A false, R true

Q1

Assertion (A): Impure method changes object state

Reason (R): Formal parameter change affects actual parameter

✓ Answer: a

💡 Explanation:

- Impure method modifies data ✓

- Hence state changes ✓

Q2

Assertion: User-defined method is written by programmer

Reason: Helps customise code

✓ Answer: a

Q3

Assertion: Method can be called without object

Reason: Static methods can be called without object

✓ Answer: a

Q4

Assertion: Recursion → method calls itself

Reason: Used for circular problems

✓ Answer: b

Q5

Assertion: All methods must return value

Reason: void methods do not return

✓ Answer: d

□ D. Case-based Questions

Q1. Output of program

Code summary:

- Circle → $\pi \times r^2 = 3.14 \times 5 \times 5 = 78.5398$
- Rectangle → $4 \times 6 = 24$
- Triangle → $0.5 \times 3 \times 4 = 6$

✓ Answer: a

Q2.

Return of areaOfRectangle(4,6)

✓ Answer: c. 24

Q3.

Object name:

Areacal cal = new Areacal();

✓ Answer: b. cal

Q4.

Variable storing triangle value:

double triangleArea

✓ Answer: c. triangleArea, double

Q5.

Purpose of String[] args

✓ Answer: a

☐ UNSOLVED PROGRAMS (Answers with Code)

Q1 Automorphic Number

```
class Auto {  
    public static void main(String[] args) {  
        int n = 5;  
        int sq = n * n;  
  
        if (sq % 10 == n)  
            System.out.println("Automorphic");  
        else  
            System.out.println("Not Automorphic");  
    }  
}
```

Q2 Calculator using switch

```
class Calc {  
    int sum(int a, int b) { return a + b; }  
    int sub(int a, int b) { return a - b; }  
    int mul(int a, int b) { return a * b; }  
}
```

Q3 Vowels & Consonants

```
class VC {
    void vowel(String s) {
        for(char c : s.toCharArray())
            if("aeiouAEIOU".indexOf(c)!=-1)
                System.out.print(c);
    }
}
```

Q4 Reverse String

```
class Rev {
    String rev(String s) {
        return new StringBuilder(s).reverse().toString();
    }
}
```

Q5 Prime Numbers

```
class Prime {
    boolean isPrime(int n) {
        for(int i=2;i<=n/2;i++)
            if(n%i==0) return false;
        return true;
    }
}
```

Q6 Area & Perimeter

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

```
class Rect {
    int area(int l, int b) {
        return l * b;
    }
}
```

```
int peri(int l, int b) {
    return 2 * (l + b);
}
```

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

```
// Input
    System.out.print("Enter length: ");
    int l = sc.nextInt();
```

```
    System.out.print("Enter breadth: ");
    int b = sc.nextInt();
```

```
// Object creation
    Rect r = new Rect();
```

```
// Method calls
int area = r.area(l, b);
int perimeter = r.peri(l, b);

// Output
System.out.println("Area = " + area);
System.out.println("Perimeter = " + perimeter);
}
}
```

● Q7.

Question:

Write a program to input time in seconds from the main() method and using method time(int) convert it into hours, minutes and seconds.

✓ Answer (Program):

```
class TimeConvert {
    void time(int sec) {
        int h = sec / 3600;
        int m = (sec % 3600) / 60;
        int s = sec % 60;

        System.out.println("Hours = " + h);
        System.out.println("Minutes = " + m);
        System.out.println("Seconds = " + s);
    }

    public static void main(String args[]) {
        TimeConvert t = new TimeConvert();
        t.time(3665);
    }
}
```

💡 Explanation:

- 1 hour = 3600 sec
- Remaining → minutes & seconds

● Q8.

Question:

Write a program to check whether a number is **Pronic number** or not using method.

✓ Answer (Program):

```
class Pronic {
    boolean check(int n) {
        for(int i=1; i<n; i++) {
            if(i * (i+1) == n)
                return true;
        }
        return false;
    }
}
```

```

public static void main(String args[]) {
    Pronic p = new Pronic();
    int num = 12;

    if(p.check(num))
        System.out.println("Pronic Number");
    else
        System.out.println("Not Pronic");
    }
}

```

💡 **Explanation:**

- $\text{Pronic} = n = x(x+1)$
- Example: $12 = 3 \times 4$ ✓

● Q9.

Question:

Using method `int GCD(int, int)` and `int LCM(int, int)` find GCD and LCM.

✓ **Answer:**

```

class GcdLcm {
    int GCD(int a, int b) {
        while(b != 0) {
            int t = b;
            b = a % b;
            a = t;
        }
        return a;
    }

    int LCM(int a, int b) {
        return (a * b) / GCD(a, b);
    }

    public static void main(String args[]) {
        GcdLcm g = new GcdLcm();
        int a = 12, b = 18;

        System.out.println("GCD = " + g.GCD(a,b));
        System.out.println("LCM = " + g.LCM(a,b));
    }
}

```

💡 **Explanation:**

- $\text{GCD} \rightarrow \text{Euclidean method}$
- $\text{LCM} = (a \times b) / \text{GCD}$

● Q10.

Question:

Write a program to calculate **tax** based on salary using method.

✓ Answer (Program):

```
class Tax {
    double cal_tax(double sal) {
        if(sal <= 100000) return 0;
        else if(sal <= 200000) return sal * 0.05;
        else if(sal <= 300000) return sal * 0.10;
        else return sal * 0.15;
    }

    public static void main(String args[]) {
        Tax t = new Tax();
        System.out.println(t.cal_tax(250000));
    }
}
```

● Q11.

Question:

Sales Discount:

Sales	Discount
Above 10000	10%
6000–9999	8%
2000–5999	4%
<2000	No discount

✓ Answer:

```
class Discount {
    void calc(double sales) {
        double dis;

        if(sales > 10000) dis = sales * 0.10;
        else if(sales >= 6000) dis = sales * 0.08;
        else if(sales >= 2000) dis = sales * 0.04;
        else dis = 0;

        System.out.println("Discount = " + dis);
    }

    public static void main(String args[]) {
        Discount d = new Discount();
        d.calc(7000);
    }
}
```

● Q12. Electricity Bill

Rates:

Units	Rate
First 100	5
Next 100	8
Next 200	10
Above 450	15

✓ Answer:

```
class Bill {
    void bill(int unit) {
        double amt = 0;

        if(unit <= 100)
            amt = unit * 5;
        else if(unit <= 200)
            amt = 100*5 + (unit-100)*8;
        else if(unit <= 450)
            amt = 100*5 + 100*8 + (unit-200)*10;
        else
            amt = 100*5 + 100*8 + 250*10 + (unit-450)*15;

        System.out.println("Bill = " + amt);
    }

    public static void main(String args[]) {
        new Bill().bill(500);
    }
}
```

● Q13. Method Overloading (Area)

✓ Answer:

```
class Area {
    void area(int l, int b) {
        System.out.println("Rectangle = " + (l*b));
    }

    void area(int a) {
        System.out.println("Square = " + (a*a));
    }

    void area(double r) {
        System.out.println("Circle = " + (3.14*r*r));
    }

    public static void main(String args[]) {
        Area a = new Area();
        a.area(5,4);
        a.area(6);
        a.area(2.5);
    }
}
```

● Q14. Overloaded Area Program

✓ Answer:

```
class Shapes {
    double area(double b, double h) {
        return b * h; // parallelogram
    }

    double area(double d1, double d2, int x) {
        return 0.5 * d1 * d2; // rhombus
    }

    double area(int a, int b, double h) {
```

```

        return 0.5 * (a + b) * h; // trapezium
    }
}

```

● Q15. Overloading (Distance / Rectangle)

✓ Answer:

```

class Shape {
    void shape(int m, int n) {
        for(int i=1;i<=m;i++) {
            for(int j=1;j<=n;j++)
                System.out.print("X");
            System.out.println();
        }
    }
}

```

● Q16. word() Overloading

✓ Answer:

```

class Word {
    String word(String w) {
        String rev = "";
        for(int i=w.length()-1;i>=0;i--)
            rev += w.charAt(i);
        return rev;
    }

    int word(String w, char ch) {
        int count = 0;
        for(int i=0;i<w.length();i++) {
            char c = w.charAt(i);
            if("aeiouAEIOU".indexOf(c) != -1)
                count++;
        }
        return count;
    }
}

```

● Q17. palindrome() Overloading

✓ Answer:

```

class Palindrome {
    void palindrome(String s) {
        String rev = new StringBuilder(s).reverse().toString();
        if(s.equalsIgnoreCase(rev))
            System.out.println("Palindrome word");
    }

    void palindrome(int n) {
        int rev=0, t=n;
        while(n>0) {
            rev = rev*10 + n%10;
            n/=10;
        }
        if(t==rev)
            System.out.println("Palindrome number");
    }
}

```

● Q18. series() Overloading

✓ Answer:

```
class Series {
    int series(int n) {
        int sum=0;
        for(int i=1;i<=n;i++)
            sum += i*i;
        return sum;
    }

    void series(int x, int n) {
        int sum=0;
        for(int i=1;i<=n;i+=2)
            sum += (x-i);
        System.out.println(sum);
    }

    void series(double n) {
        for(int i=1;i<=n;i++)
            System.out.print(i*i + " ");
    }
}
```

● Q19. show() Overloading

✓ Answer:

```
class Show {
    double show(double n) {
        double sum=0;
        for(int i=1;i<=n;i++)
            sum += (i*i)/(i*4);
        return sum;
    }

    double show(double a, double n) {
        double sum=0;
        for(int i=1;i<=n;i++)
            sum += (i*a)/(i*5);
        return sum;
    }
}
```
