

Class 10

Chapter – 11

Class as the Basis of All Computation

A. TICK (✓) THE CORRECT ANSWER

● A. Tick (✓) the correct answer

Q1.

Which keyword is used to create an instance of a class in many object-oriented languages?

- a. def
- b. class
- c. new
- d. create

✓ Answer: c. new

💡 Explanation:

- `new` is used to create an object in Java

```
Employee e = new Employee();
```

Q2.

Which of the following is NOT a feature of object-oriented programming?

- a. Encapsulation
- b. Polymorphism
- c. Abstraction
- d. Compilation

✓ Answer: d. Compilation

💡 Explanation:

- OOP features = Encapsulation, Inheritance, Polymorphism, Abstraction
 - Compilation is NOT a feature
-

Q3.

What does polymorphism allow in object-oriented programming?

- a. Multiple classes to inherit from one class
- b. Methods to operate on different types of objects
- c. A class to have more than one instance
- d. Data hiding from external classes

✓ Answer: b

💡 Explanation:

- Same method name → different behaviour
Example: method overloading
-

Q4.

Which of the following best describes a constructor in a class?

- a. A method for destroying objects
- b. A method for creating and initializing objects
- c. A method that can be called multiple times
- d. A method that prevents inheritance

✓ Answer: b

💡 Explanation:

- Constructor initializes object

```
Employee() {  
    basic = 0;  
}
```

Q5.

What is abstraction in object-oriented programming?

- a. Ability of different classes to be treated as same
- b. Hiding complex implementation and showing only essential features
- c. Using pointers
- d. Converting one object to another

✓ Answer: b

💡 Explanation:

- Abstraction = Hide details, show only important parts
-

B. Answer the following questions

● Q1. Employee Class Program

✓ Question Summary:

Create class `Employee` with:

- Name, Basic
 - DA = 25%
 - HRA = 15%
 - PF = 8.33%
 - Net Pay = Basic + DA + HRA
 - Gross Pay = Net Pay – PF
-

✓ Program:

```
import java.util.*;

class Employee {
    String name;
    double basic, da, hra, pf, np, gp;

    void input() {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter Name: ");
        name = sc.nextLine();

        System.out.print("Enter Basic Salary: ");
        basic = sc.nextDouble();
    }

    void calc() {
        da = 0.25 * basic;
        hra = 0.15 * basic;
        pf = 0.0833 * basic;

        np = basic + da + hra;
        gp = np - pf;
    }

    void display() {
        System.out.println("\nName: " + name);
        System.out.println("Basic: " + basic);
        System.out.println("DA: " + da);
        System.out.println("HRA: " + hra);
        System.out.println("PF: " + pf);
        System.out.println("Net Pay: " + np);
        System.out.println("Gross Pay: " + gp);
    }

    public static void main(String args[]) {
        Employee e = new Employee();
    }
}
```

```
e.input();
e.calc();
e.display();
}
}
```

💡 Explanation (Simple):

- DA = 25% of basic
- HRA = 15%
- PF = deduction
- Net Pay = salary before deduction
- Gross Pay = final salary

🌐 Q2. Tax Class Program

✓ Question Summary:

Class tax:

- Name, Department, Monthly Salary, Income Tax
- Tax = 12% if annual salary > 2,00,000

✓ Program:

```
import java.util.*;

class tax {
    String name, dept;
    double monthlySalary, incomeTax;

    void input() {
        Scanner sc = new Scanner(System.in);

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

        System.out.print("Enter Department: ");
        dept = sc.nextLine();

        System.out.print("Enter Monthly Salary: ");
        monthlySalary = sc.nextDouble();
    }

    void compute() {
        double annual = monthlySalary * 12;

        if(annual > 200000)
            incomeTax = annual * 0.12;
        else
```

```

        incomeTax = 0;
    }

    void display() {
        System.out.println("\nName: " + name);
        System.out.println("Department: " + dept);
        System.out.println("Monthly Salary: " + monthlySalary);
        System.out.println("Income Tax: " + incomeTax);
    }

    public static void main(String args[]) {
        tax t = new tax();
        t.input();
        t.compute();
        t.display();
    }
}

```

💡 Explanation:

- Annual = monthly × 12
- Tax only if > 2,00,000
- Else → 0

🎯 Q3. Class Bill

✓ Program:

```

import java.util.*;

class Bill {
    String name;
    int cons_no;
    int unit_con;
    double bill;

    void getdata() {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter Name: ");
        name = sc.nextLine();
        System.out.print("Enter Consumer No: ");
        cons_no = sc.nextInt();
        System.out.print("Enter Units Consumed: ");
        unit_con = sc.nextInt();
    }

    void compute() {
        if(unit_con <= 120)
            bill = unit_con * 1.20;
        else if(unit_con <= 250)
            bill = unit_con * 2.20;
        else if(unit_con <= 400)
            bill = unit_con * 3.20;
        else
            bill = unit_con * 4.00;
    }

    void display() {
        System.out.println("\nConsumer Name: " + name);
        System.out.println("Consumer No: " + cons_no);
    }
}

```

```

        System.out.println("Units: " + unit_con);
        System.out.println("Bill Amount: " + bill);
    }

    public static void main(String args[]) {
        Bill b = new Bill();
        b.getdata();
        b.compute();
        b.display();
    }
}

```

💡 Explanation:

- Units → decide rate
- Multiply → get bill

🌐 Q4. Class School_book

✓ Program:

```

import java.util.*;

class School_book {
    String name, author_name;
    double price_book;
    int number_book, total;

    void accept() {
        Scanner sc = new Scanner(System.in);
        System.out.print("Book Name: ");
        name = sc.nextLine();
        System.out.print("Author Name: ");
        author_name = sc.nextLine();
        System.out.print("Price: ");
        price_book = sc.nextDouble();
        System.out.print("Number of Books: ");
        number_book = sc.nextInt();
    }

    void cal_cost() {
        total = (int)(price_book * number_book);
    }

    void display() {
        System.out.println("\nBook: " + name);
        System.out.println("Author: " + author_name);
        System.out.println("Total Cost: " + total);
    }

    public static void main(String args[]) {
        School_book s = new School_book();
        s.accept();
        s.cal_cost();
        s.display();
    }
}

```

● Q5. Class BookFair

✓ Program:

```
import java.util.*;

class BookFair {
    String Bname;
    double price, discount, finalPrice;

    void input() {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter Book Name: ");
        Bname = sc.nextLine();
        System.out.print("Enter Price: ");
        price = sc.nextDouble();
    }

    void calculate() {
        if(price <= 1000)
            discount = price * 0.02;
        else if(price <= 3000)
            discount = price * 0.10;
        else
            discount = price * 0.15;

        finalPrice = price - discount;
    }

    void display() {
        System.out.println("\nBook: " + Bname);
        System.out.println("Final Price: " + finalPrice);
    }

    public static void main(String args[]) {
        BookFair b = new BookFair();
        b.input();
        b.calculate();
        b.display();
    }
}
```

● Q6. Class RailwayTicket

✓ Program:

```
import java.util.*;

class RailwayTicket {
    String name, coach;
    long mobno;
    int amt, totalamt;

    void accept() {
        Scanner sc = new Scanner(System.in);
        System.out.print("Name: ");
        name = sc.nextLine();
        System.out.print("Coach: ");
        coach = sc.nextLine();
        System.out.print("Mobile No: ");
        mobno = sc.nextLong();
    }
}
```

```

        System.out.print("Amount: ");
        amt = sc.nextInt();
    }

    void update() {
        if(coach.equalsIgnoreCase("First_AC"))
            totalamt = amt + 700;
        else if(coach.equalsIgnoreCase("Second_AC"))
            totalamt = amt + 500;
        else if(coach.equalsIgnoreCase("Third_AC"))
            totalamt = amt + 250;
        else
            totalamt = amt;
    }

    void display() {
        System.out.println("\nName: " + name);
        System.out.println("Coach: " + coach);
        System.out.println("Total Amount: " + totalamt);
        System.out.println("Mobile: " + mobno);
    }

    public static void main(String args[]) {
        RailwayTicket r = new RailwayTicket();
        r.accept();
        r.update();
        r.display();
    }
}

```

Q7. Class Discount

Program:

```

import java.util.*;

class Discount {
    String item;
    double cost, dis, amt;

    void input() {
        Scanner sc = new Scanner(System.in);
        System.out.print("Item Name: ");
        item = sc.nextLine();
        System.out.print("Price: ");
        cost = sc.nextDouble();
    }

    void cal() {
        if(cost <= 2000)
            dis = 0;
        else if(cost <= 5000)
            dis = cost * 0.10;
        else if(cost <= 7000)
            dis = cost * 0.15;
        else
            dis = cost * 0.20;

        amt = cost - dis;
    }
}

```

```

    void display() {
        System.out.println("\nItem: " + item);
        System.out.println("Price: " + cost);
        System.out.println("Discount: " + dis);
        System.out.println("Amount: " + amt);
    }

    public static void main(String args[]) {
        Discount d = new Discount();
        d.input();
        d.cal();
        d.display();
    }
}

```

Q8. Class Interest

✓ Program:

```

import java.util.*;

class interest {
    double p, r, si, amt;
    int t;

    void input() {
        Scanner sc = new Scanner(System.in);
        System.out.print("Principal: ");
        p = sc.nextDouble();
        System.out.print("Rate: ");
        r = sc.nextDouble();
        System.out.print("Time: ");
        t = sc.nextInt();
    }

    void calculate() {
        si = (p * r * t) / 100;
        amt = p + si;
    }

    void show() {
        System.out.println("Simple Interest: " + si);
        System.out.println("Amount: " + amt);
    }

    public static void main(String args[]) {
        interest i = new interest();
        i.input();
        i.calculate();
        i.show();
    }
}

```

Q9. Class School_Library

✓ Program:

```

import java.util.*;

class School_Library {

```

```

String name;
int day_late;
double fine;

void accept() {
    Scanner sc = new Scanner(System.in);
    System.out.print("Book Name: ");
    name = sc.nextLine();
    System.out.print("Days Late: ");
    day_late = sc.nextInt();
}

void calculate() {
    if(day_late <= 5)
        fine = day_late * 5;
    else if(day_late <= 15)
        fine = day_late * 7;
    else if(day_late <= 30)
        fine = day_late * 10;
    else
        fine = day_late * 15;
}

void display() {
    System.out.println("\nBook: " + name);
    System.out.println("Days Late: " + day_late);
    System.out.println("Fine: " + fine);
}

public static void main(String args[]) {
    School_Library s = new School_Library();
    s.accept();
    s.calculate();
    s.display();
}
}

```

Q10. Loan Interest Program

✓ Question (as per book):

Write a program to calculate interest based on time:

- 1 year → 4.5%
- 2 years → 5.5%
- 3 years → 7.5%

✓ Program:

```

import java.util.*;

class Loan {
    int time;
    double principal, interest, rate;

    void input() {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter Principal: ");
        principal = sc.nextDouble();
    }
}

```

```

        System.out.print("Enter Time (years): ");
        time = sc.nextInt();
    }

    void calculate() {
        if(time == 1)
            rate = 4.5;
        else if(time == 2)
            rate = 5.5;
        else if(time == 3)
            rate = 7.5;

        interest = (principal * rate * time) / 100;
    }

    void display() {
        System.out.println("Interest = " + interest);
    }

    public static void main(String args[]) {
        Loan l = new Loan();
        l.input();
        l.calculate();
        l.display();
    }
}

```

💡 Explanation:

- Rate depends on time
- Use simple interest formula

📖 Q11. Automorphic Number

✓ Program:

```

import java.util.*;

class Automorphic {
    int n, c = 0;

    void accept() {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter number: ");
        n = sc.nextInt();
    }

    int count() {
        int temp = n;
        while(temp > 0) {
            c++;
            temp /= 10;
        }
        return c;
    }

    void display() {
        int sq = n * n;
        int mod = (int)Math.pow(10, count());
    }
}

```

```

        if(sq % mod == n)
            System.out.println("Automorphic Number");
        else
            System.out.println("Not Automorphic");
    }

    public static void main(String args[]) {
        Automorphic a = new Automorphic();
        a.accept();
        a.display();
    }
}

```

💡 Explanation:

Example: $25 \rightarrow 25^2 = 625 \rightarrow$ ends with 25 ✓

📖 Q12. Palindrome between m and n

✓ Program:

```

import java.util.*;

class PalindromeRange {
    int m, n;

    void accept() {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter m and n: ");
        m = sc.nextInt();
        n = sc.nextInt();
    }

    int palindrome(int num) {
        int rev = 0, temp = num;
        while(temp > 0) {
            rev = rev * 10 + temp % 10;
            temp /= 10;
        }
        return rev;
    }

    void display() {
        for(int i = m; i <= n; i++) {
            if(i == palindrome(i))
                System.out.print(i + " ");
        }
    }

    public static void main(String args[]) {
        PalindromeRange p = new PalindromeRange();
        p.accept();
        p.display();
    }
}

```

Q13. Quadratic Equation

✓ Program:

```
import java.util.*;

class Quadratic {
    double a, b, c, d, r1, r2;

    void accept() {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter a, b, c: ");
        a = sc.nextDouble();
        b = sc.nextDouble();
        c = sc.nextDouble();
    }

    void compute() {
        d = b*b - 4*a*c;

        if(d >= 0) {
            r1 = (-b + Math.sqrt(d)) / (2*a);
            r2 = (-b - Math.sqrt(d)) / (2*a);
            System.out.println("Roots: " + r1 + ", " + r2);
        } else {
            System.out.println("Imaginary roots");
        }
    }

    public static void main(String args[]) {
        Quadratic q = new Quadratic();
        q.accept();
        q.compute();
    }
}
```

Q14. Reverse String

✓ Program:

```
import java.util.*;

class ReverseStr {
    String str, rev = "";

    void accept() {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter sentence: ");
        str = sc.nextLine();
    }

    void reverse() {
        for(int i = str.length()-1; i >= 0; i--)
            rev += str.charAt(i);
    }

    void show() {
        System.out.println("Reversed: " + rev);
    }

    public static void main(String args[]) {
```

```
        ReverseStr r = new ReverseStr();
        r.accept();
        r.reverse();
        r.show();
    }
}
```

Q15. Count Vowels & Consonants

✓ Program:

```
import java.util.*;

class Word {
    String w;
    int vc = 0, cc = 0;

    void input() {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter word: ");
        w = sc.nextLine().toLowerCase();
    }

    void count_vowels() {
        for(int i=0; i<w.length(); i++) {
            char ch = w.charAt(i);
            if("aeiou".indexOf(ch) != -1)
                vc++;
            else if(Character.isLetter(ch))
                cc++;
        }
    }

    void display() {
        System.out.println("Vowels: " + vc);
        System.out.println("Consonants: " + cc);
    }

    public static void main(String args[]) {
        Word obj = new Word();
        obj.input();
        obj.count_vowels();
        obj.display();
    }
}
```

SECTION C: Assertion–Reason

Options:

- a. Both true + correct explanation
 - b. Both true but not explanation
 - c. A true, R false
 - d. A false, R true
-

✓ Q1

✓ A: True

✓ R: False

☞ **Answer: c**

✓ Q2

✓ A: True

✓ R: True (correct explanation)

☞ **Answer: a**

✓ Q3

✓ A: True

✓ R: True (correct explanation)

☞ **Answer: a**

✓ Q4

✓ A: True

✓ R: True (correct explanation)

☞ **Answer: a**

SECTION D: Case-Based Question

✓ Key Points:

Q1. `this` keyword

☞ Refers to **current object**

✓ Answer: **b**

Q2. `displayDetails()` method

☞ It prints title, author, price

✓ Answer: **b**

Q3. `this.price -= discount`

☞ Reduces price

✓ Answer: **b (decreases amount)**

Q4. Parameter type of discount

☞ double discount → formal parameter

✓ Answer: **b**

Q5. Static variable

☞ Shared among all objects

✓ Answer: **c**
