LAB RECORD

23CSE111- Object Oriented Programming

Submitted by

CH.SC.U4CSE24153 -B Lohith Krishnan

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE AND ENGINEERING

AMRITA VISHWA VIDYAPEETHAM

AMRITA SCHOOL OF COMPUTING

CHENNAI

March - 2025

AMRITA VISHWA VIDYAPEETHAM

AMRITA SCHOOL OF COMPUTING, CHENNAI

BONAFIDE CERTIFICATE

This is to certify that the Lab Record work for 23CSE111- Object Oriented Programming Subject submitted by ***CH.SC.U4CSE24153 – B LOHITH KRISHNAN*** in **"Computer Science and Engineering"** is a Bonafide record of the work carried out under my guidance and supervision at Amrita School of Computing, Chennai.

This Lab examination held on

Internal Examiner 1 Internal Examiner 2

INDEX

S.NO	TITLE	PAGE.NO
	UML DIAGRAM	
1.	**COURSE MANAGCOURSEEMENT SYSTEM**	
	1.a) Use Case Diagram	4
	1.b) Class Diagram	5
	1.c) Sequence Diagram	5
	1.d) Communication Diagram	7
	1.e) State-Activity Diagram	8
2.	**ONLINE SHOPPING SYSTEM**	
	2.a) Use Case Diagram	9
	2.b) Class Diagram	10
	2.c) Sequence Diagram	10
	2.d) Communication Diagram	11
	2.e) State-Activity Diagram	11
3.	**BASIC JAVA PROGRAMS**	
	3.a) SumOfNaturalNumbers	12
	3.b) Factorial	13
	3.c) Fibonacci	14
	3.d) ReversedNumber	14
	3.e) CheckPalindrome	15
	3.f) OddOrEven	16
	3.g) SquareSum	16
	3.h) CheckPerfectSquare	17
	3.i) AreaOfTriangle	18
	3.j) PositveOrNegative	18
4.	**INHERITANCE**	
	SINGLE INHERITANCE PROGRAMS	
	4.a)Bank customer details	22

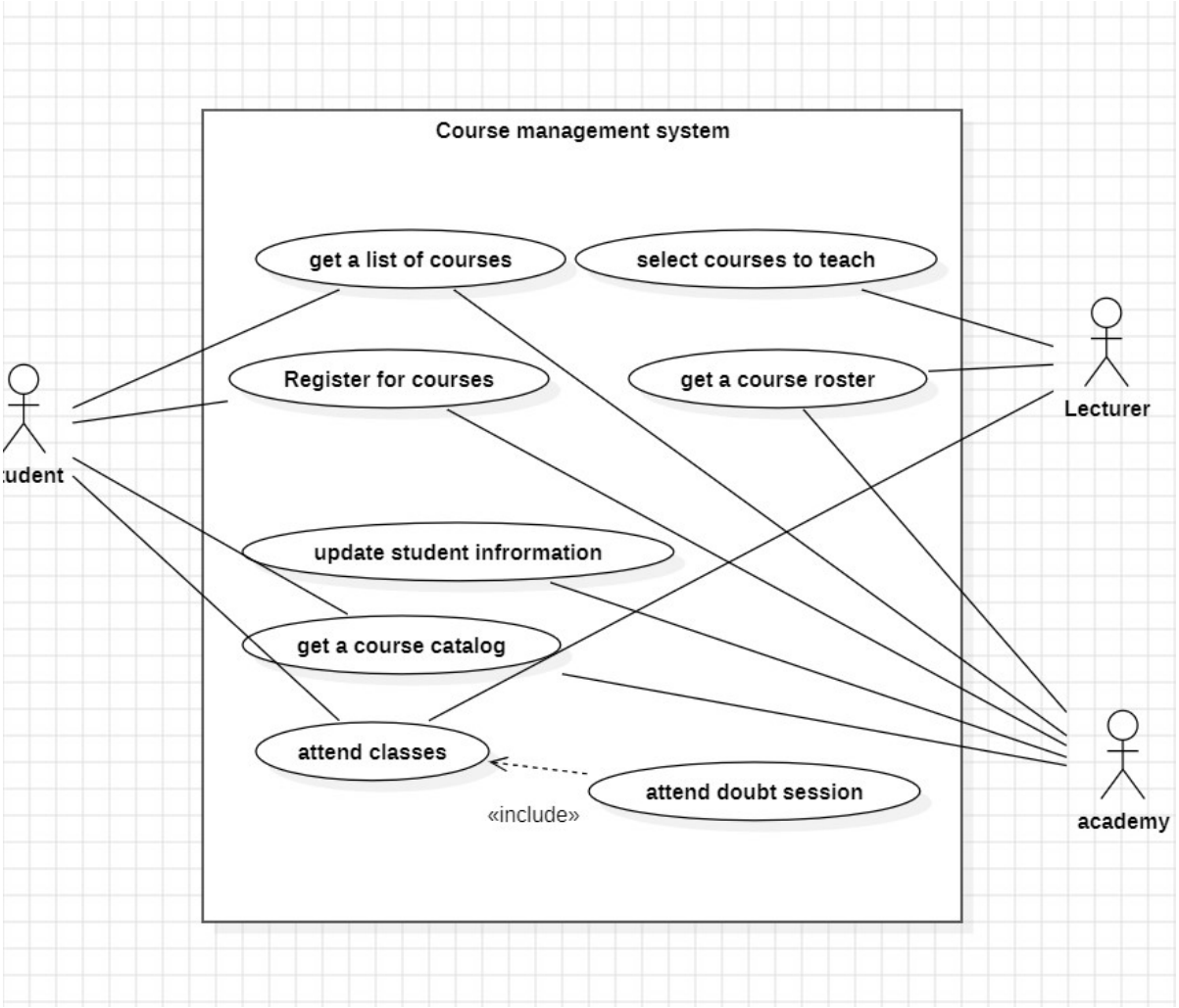
	4.b)Employee salary details	23
5.	**MULTILEVEL INHERITANCE PROGRAMS**	
	5.a)Company details	25
	5.b)University student grade	26
6.	**HIERARCHICAL INHERITANCE PROGRAMS**	
	6.a)Payment	28
	6.b)User log in	29
7.	**HYBRID INHERITANCE PROGRAMS**	
	7.a)Vehicle	31
	7.b)Gadget	32
.	**POLYMORPHISM**	
8.	**CONSTRUCTOR PROGRAMS**	
	8.a)Movie	34
9.	**CONSTRUCTOR OVERLOADING PROGRAMS**	
	9.a)Book details	35
10.	**METHOD OVERLOADING PROGRAMS**	
	10.a)Calculate area	37
	10.b)Unit converter	37
11.	**METHOD OVERRIDING PROGRAMS**	
	11.a)Area	39
	11.b)Languages	40
	ABSTRACTION	
12.	**INTERFACE PROGRAMS**	
	12.a)Math operations	42
	12.b)Temperature converter	43
	12.c)College	44
	12.d)String operations	45
13.	**ABSTRACT CLASS PROGRAMS**	
	13.a)Account	47
	13.b)Media player	48
	13.c)Report details	49

	13.d)Editor	50
	ENCAPSULATION	
14.	**ENCAPSULATION PROGRAMS**	
	14.a)Movie details	52
	14.b)Hotel booking	53
	14.c)Loan interest	54
	14.d)Rectangle	55
15.	**PACKAGES PROGRAMS**	
	15.a)Exponents	56
	15.b)Temperature converter	56
	15.c)Sort even numbers	57
	15.d)Date	58
16.	**EXCEPTION HANDLING PROGRAMS**	
	16.a)Check input format	59
	16.b)Password check	60
	16.c)Email validation	61
	16.d)Age check	62
17.	**FILE HANDLING PROGRAMS**	
	17.a)Deleting file	63
	17.b)Get file details	64
	17.c)Create new file	65
	17.d)Write in a new file	66

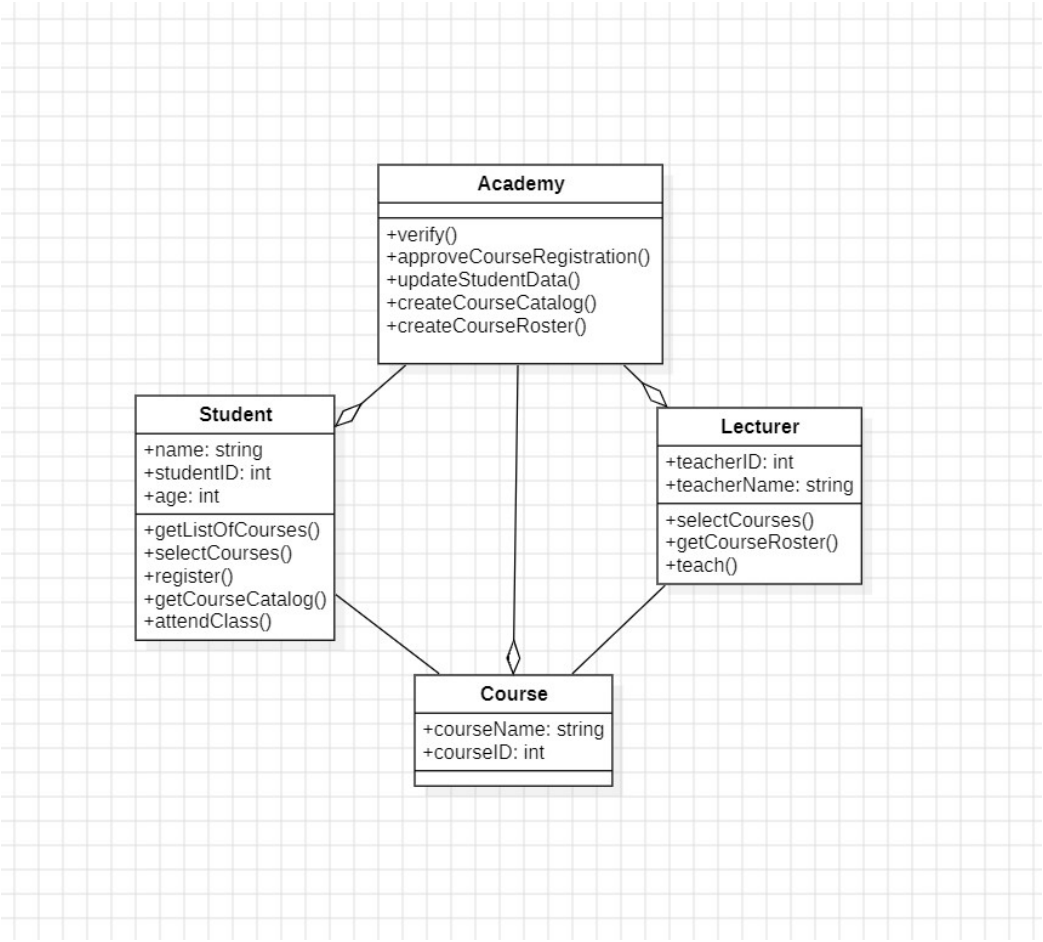
UML DIAGRAMS

1. COURSE MANAGEMENT SYSTEM

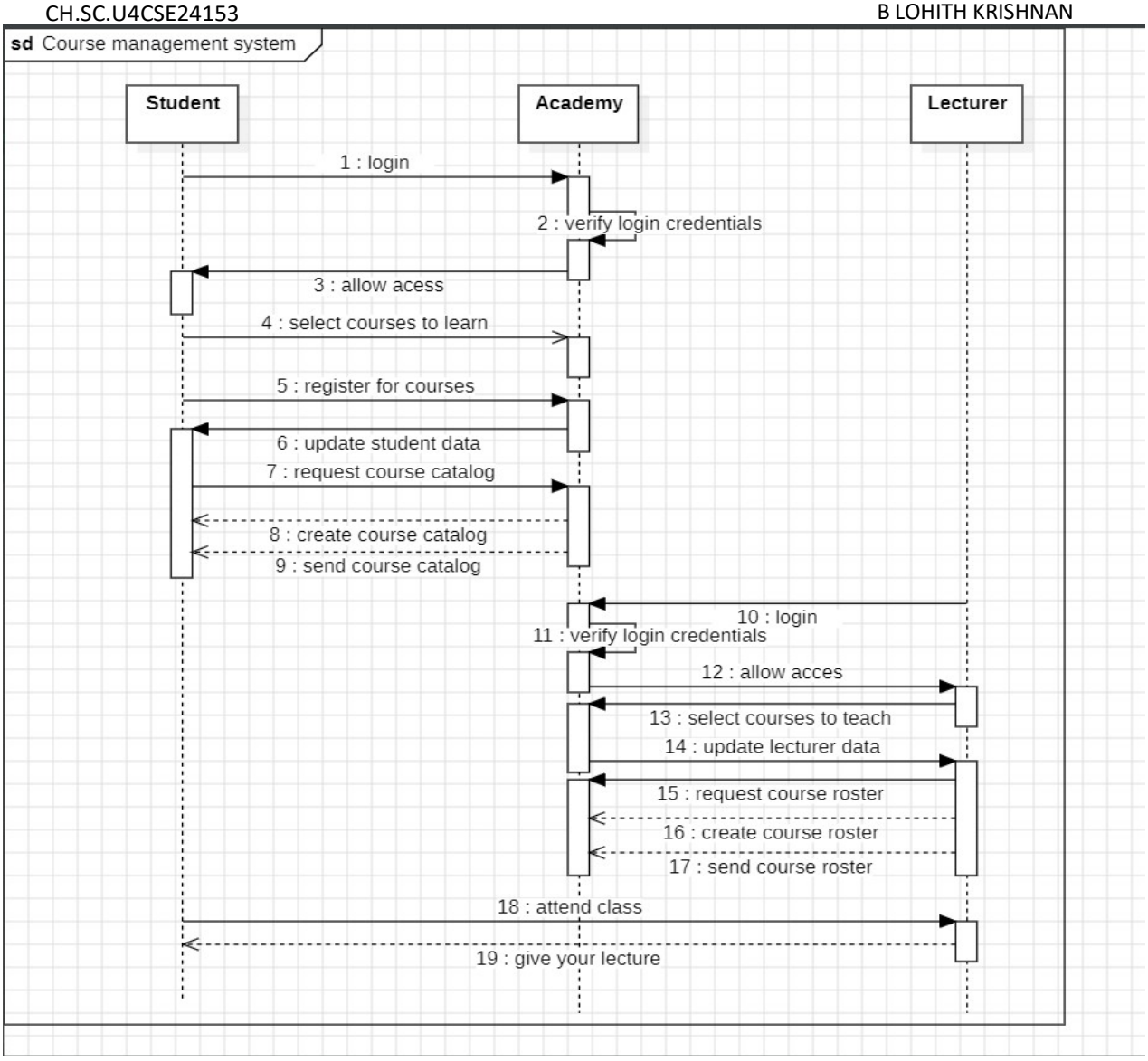
1.a) Use Case Diagram:



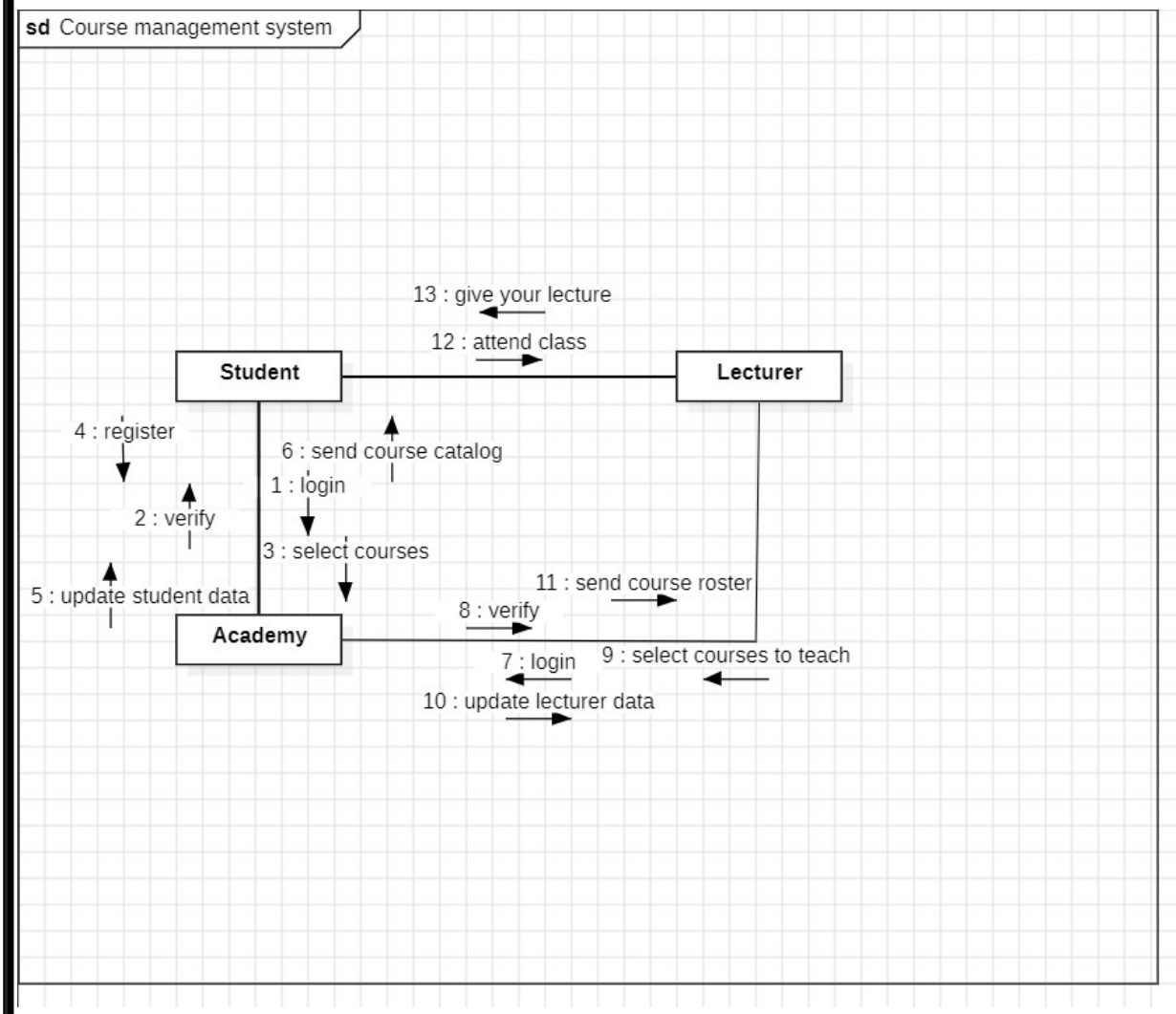
1.b) Class Diagram:



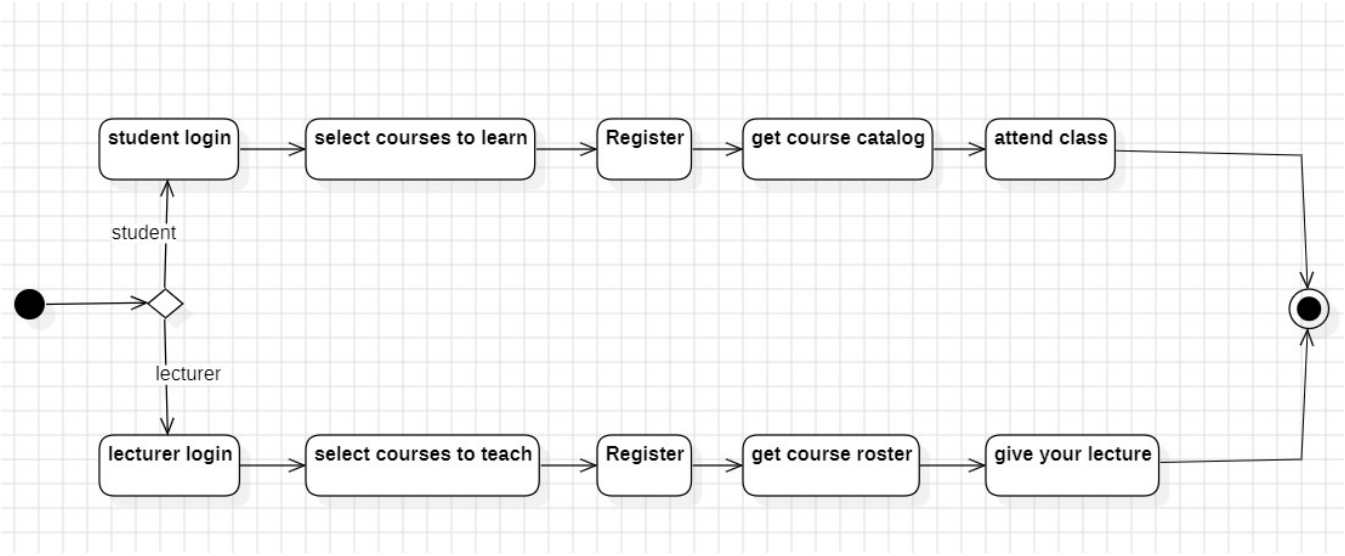
1.c) Sequence Diagram:


sd Course management system
Student
Academy
Lecturer
1 : login
2 : verify login credentials
3 : allow acess
4 : select courses to learn
5 : register for courses
6 : update student data
7 : request course catalog
8 : create course catalog
9 : send course catalog
10 : login
11 : verify login credentials
12 : allow acces
13 : select courses to teach
14 : update lecturer data
15 : request course roster
16 : create course roster
17 : send course roster
18 : attend class
19 : give your lecture

1.d) Communication Diagram:

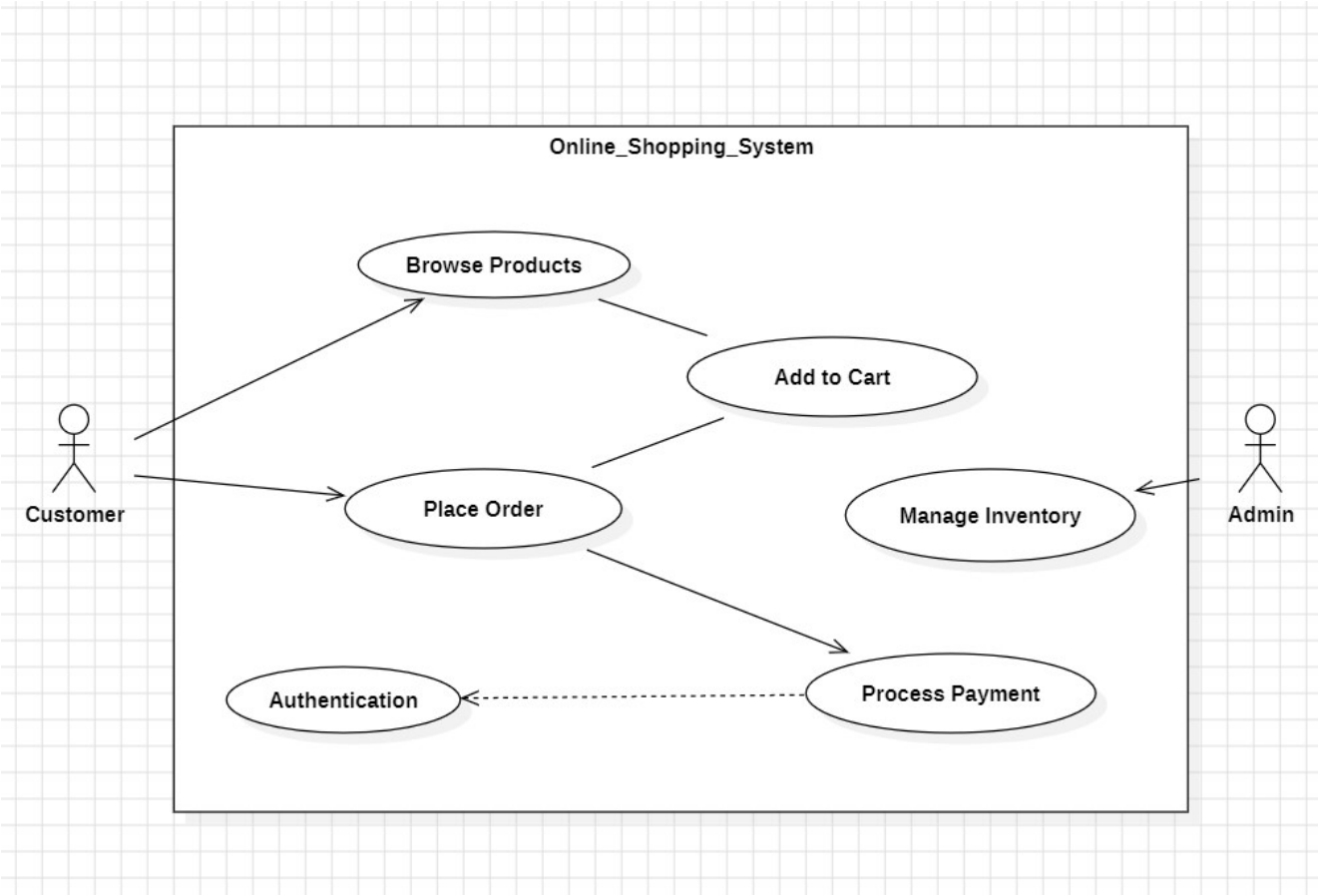


1.e) State-Activity Diagram:

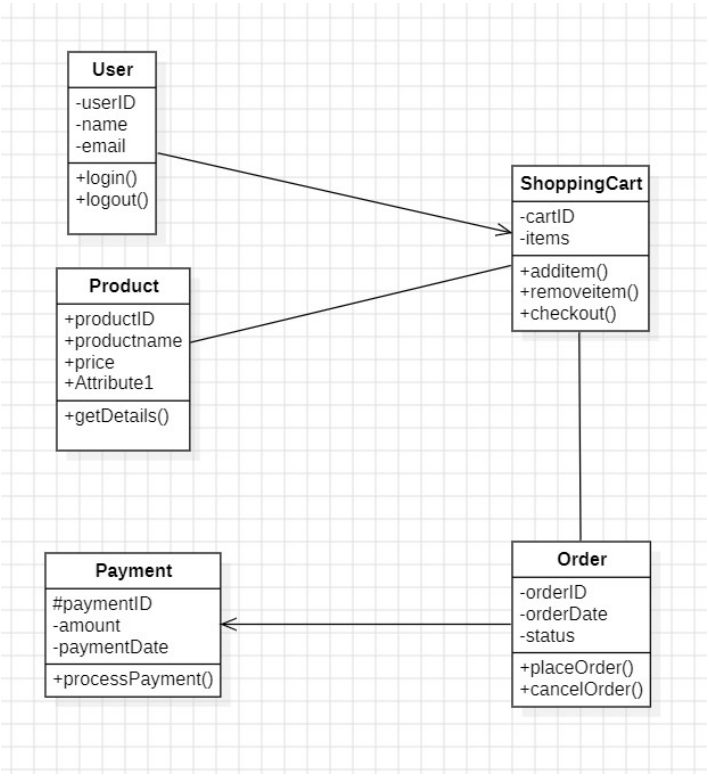


2. ONLINE SH0PPING SYSTEM

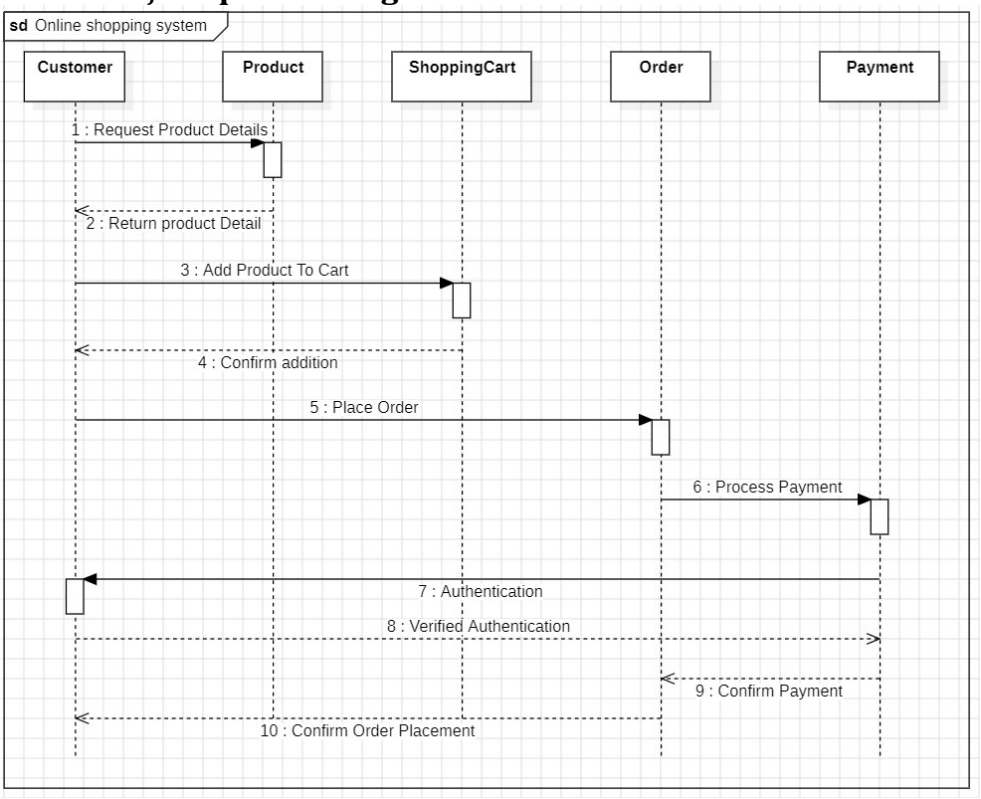
2.a) Use Case Diagram:



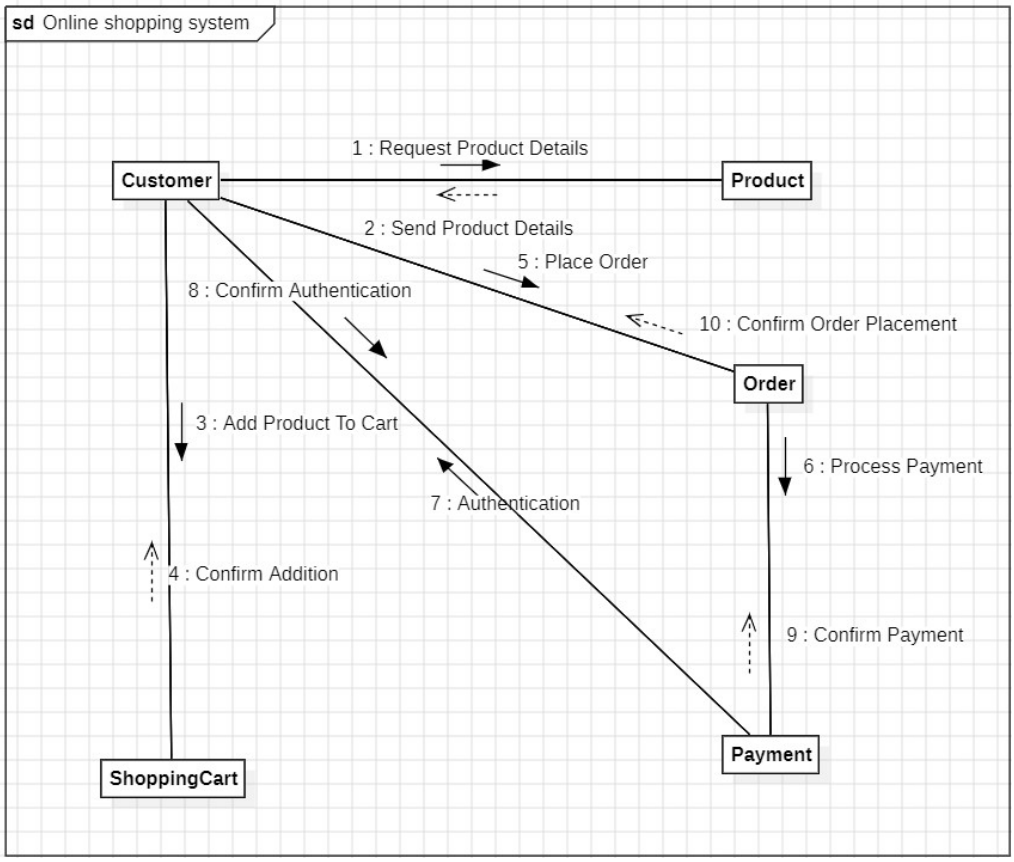
2.b) Class Diagram:



2.c) Sequence Diagram:



2.d) Communication Diagram:



2.e) State-Activity Diagram:

3. Basic Java Programs

3.a) SumOfNaturalNumbers:

Code:

```
import java.util.Scanner;

public class SumOfNaturalNumbers {

    public static void main(String[] args) {

        Scanner scanner = new Scanner(System.in);

        System.out.print("Enter a positive integer: ");

        int n = scanner.nextInt();
        scanner.close();

        int sum = 0;

        for (int i = 1; i <= n; i++) {
            sum += i;
        }

        System.out.println("The sum of the first " + n + " natural numbers is " +
sum);
    }
}
```

Output:

```
C:\Users\lohit\OneDrive\Documents\opps program>javac SumOfNaturalNumbers.java

C:\Users\lohit\OneDrive\Documents\opps program>java SumOfNaturalNumbers
Enter a positive integer: 8
The sum of the first 8 natural numbers is 36

C:\Users\lohit\OneDrive\Documents\opps program>
```

3.b) factorial :

Code:

```
import java.util.Scanner;

public class factorial {

    public static void main(String[] args) {
            Scanner scanner = new Scanner(System.in);
            System.out.print("Enter a positive integer: ");
            int n = scanner.nextInt();
            int fact = 1;
      for (int i = 1; i <= n; i++) {
        fact = fact * i;
        }
      System.out.println("Factorial: "+ fact);
    }
}
```

Output:

```
C:\Users\lohit\OneDrive\Documents\opps program>javac factorial.java

C:\Users\lohit\OneDrive\Documents\opps program>java factorial
Enter a positive integer: 5
Factorial: 120

C:\Users\lohit\OneDrive\Documents\opps program>
```

3.c) Fibonacci:

Code:

```
import java.util.Scanner;

    public class fibonacci {
      public static void main(String[] args) {
      Scanner scanner = new Scanner(System.in);
      System.out.print("Enter a positive integer: ");
      int n=scanner.nextInt();
      int a = 0;
      int b=1;
      System.out.print(a+" "+b);
      for (int i=2; i<n; i++) {
      int next = a + b;
      System.out.print(" " + next);
      a=b;
      b=next;
      }
    }
}
```

Output:

```
C:\Users\lohit\OneDrive\Documents\opps program>javac fibonacci.java

C:\Users\lohit\OneDrive\Documents\opps program>java fibonacci
Enter a positive integer: 9
0 1 1 2 3 5 8 13 21
C:\Users\lohit\OneDrive\Documents\opps program>
```

3.d) ReversedNumber:

Code:

```
import java.util.Scanner;

    public class ReversedNumber {
      public static void main(String[] args) {
      Scanner scanner = new Scanner(System.in);
      System.out.print("Enter a positive integer: ");
      int n = scanner.nextInt();
      int reversed = 0;
```

```
      while (n != 0) {
      int digit = n % 10;
      reversed = reversed * 10 + digit;
      n /= 10;
      }
      System.out.println("Reversed Number: " + reversed);
     }
}
```

Output:

```
C:\Users\lohit\OneDrive\Documents\opps program>javac ReversedNumber.java

C:\Users\lohit\OneDrive\Documents\opps program>java ReversedNumber
Enter a positive integer: 53278
Reversed Number: 87235

C:\Users\lohit\OneDrive\Documents\opps program>
```

3.e) CheckPalindrome:

Code:

```
import java.util.Scanner;

   public class CheckPalindrome {
      public static boolean isPalindrome(String s) {
      s = s.toLowerCase();
      String rev = "";
      for (int i = s.length() -1; i >= 0; i--) {
      rev = rev + s.charAt(i);
      }
      return s.equals(rev);
      }

      public static void main(String[] args) {
      Scanner scanner = new Scanner(System.in);
      System.out.print("Enter a word: ");
      String s = scanner.nextLine();
      boolean res = isPalindrome(s);
      if (res) {
      System.out.println("\"" + s + "\" is a palindrome.");
      } else {
      System.out.println("\"" + s + "\" is not a palindrome.");
      }
   }
}
```

Output:

```
C:\Users\lohit\OneDrive\Documents\opps program>javac CheckPalindrome.java

C:\Users\lohit\OneDrive\Documents\opps program>java CheckPalindrome
Enter a word: racecar
"racecar" is a palindrome.

C:\Users\lohit\OneDrive\Documents\opps program>
```

3.f) OddOrEven:

Code:

```
import java.util.Scanner;

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

   if (n % 2 == 0) {
   System.out.println("Entered Number is Even");
   }
   else {
   System.out.println("Entered Number is Odd");
   }
   }
}
```

Output:

```
C:\Users\lohit\OneDrive\Documents\opps program>javac OddOrEven.java

C:\Users\lohit\OneDrive\Documents\opps program>java OddOrEven
Enter a integer: 7
Entered Number is Odd

C:\Users\lohit\OneDrive\Documents\opps program>
```

3.g) SquareSum:

Code:

```
import java.util.Scanner;

public class squaresum {
   public static int squaresum(int n) {
   int sum = 0;
   for (int i = 1; i <= n; i++)
   sum += (i* i);
```

```
    return sum;
    }

    public static void main(String args[]) {
    Scanner scanner = new Scanner(System.in);
    System.out.print("Enter a integer: ");
    int n = scanner.nextInt();
    System.out.println(squaresum(n));
    }
}
```

Output:

```
C:\Users\lohit\OneDrive\Documents\opps program>javac squaresum.java

C:\Users\lohit\OneDrive\Documents\opps program>java squaresum
Enter a integer: 8
204

C:\Users\lohit\OneDrive\Documents\opps program>
```

3.h) CheckPerfectSquare:

Code:

```
import java.util.Scanner;

   public class CheckPerfectSquare {
      public static void main(String[] args) {
      Scanner scanner = new Scanner(System.in);
      System.out.println("Enter an integer:");
      int number = scanner.nextInt();
      int sqrt = (int) Math.sqrt(number);

      if (sqrt*sqrt == number) {
      System.out.println(number + " is a perfect square.");
      }
      else {
      System.out.println(number + " is not a perfect square.");
      }
   }
}
```

Output:

```
C:\Users\lohit\OneDrive\Documents\opps program>javac CheckPerfectSquare.java

C:\Users\lohit\OneDrive\Documents\opps program>java CheckPerfectSquare
Enter an integer:
64
64 is a perfect square.

C:\Users\lohit\OneDrive\Documents\opps program>
```

3.i) AreaOfTriangle:

Code:

```
import java.util.Scanner;

public class AreaOfTriangle {
   static double area (double h, double b) {
   return (h*b)/2;
   }

   public static void main(String[] args) {
   Scanner scanner = new Scanner(System.in);
   System.out.println("Enter height:");
   double h = scanner.nextDouble();
   System.out.println("Enter breadth:");
   double b = scanner.nextDouble();
   System.out.println("Area of the triangle: "+ area(h,b));
   }
}
```

Output:

```
C:\Users\lohit\OneDrive\Documents\opps program>javac AreaOfTriangle.java

C:\Users\lohit\OneDrive\Documents\opps program>java AreaOfTriangle
Enter height:
8
Enter breadth:
4
Area of the triangle: 16.0

C:\Users\lohit\OneDrive\Documents\opps program>
```

3.j) PositveOrNegative:

Code:

```
import java.util.Scanner;

public class PositveOrNegative {
   public static void main(String[] args) {
      Scanner scanner = new Scanner(System.in);
      System.out.println("Enter an integer:");
      int number = scanner.nextInt();

      if (number > 0) {
      System.out.println(number + " is positive.");
      }
      else if (number < 0) {
      System.out.println(number + " is negative.");
      }
      else {
      System.out.println(number + " is zero.");
      }
   }
}
```

Output:

```
C:\Users\lohit\OneDrive\Documents\opps program>javac PositveOrNegative.java

C:\Users\lohit\OneDrive\Documents\opps program>java PositveOrNegative
Enter an integer:
-5
-5 is negative.

C:\Users\lohit\OneDrive\Documents\opps program>
```

Inheritance

4. Single inheritance

4.a) Bank customer details

Code:

```
class Bank {
    String bankName = "SBI";
    String branch = "Mumbai";

    void showBankDetails() {
        System.out.println("Bank Name: " + bankName);
        System.out.println("Branch: " + branch);
    }
}

class Account extends Bank {
    String accountHolder;
    int accountNumber;

    Account(String accountHolder, int accountNumber) {
        this.accountHolder = accountHolder;
        this.accountNumber = accountNumber;
    }

    void showAccountDetails() {
        System.out.println("Account Holder: " + accountHolder);
        System.out.println("Account Number: " + accountNumber);
    }
}

public class Singleinheritance1 {
    public static void main(String[] args) {
        Account acc1 = new Account("Arjun", 567890);
        acc1.showBankDetails();
        acc1.showAccountDetails();
    }
}
```

Output:

```
C:\Users\lohit\Downloads\Oops Report\single inheritance>javac Singleinheritance1.java

C:\Users\lohit\Downloads\Oops Report\single inheritance>java Singleinheritance1
Bank Name: SBI
Branch: Mumbai
Account Holder: Arjun
Account Number: 567890
```

4.b) Employee salary details:

Code:

```
class Employee {
    String employeeName;
    double baseSalary;

    Employee(String employeeName, double baseSalary) {
        this.employeeName = employeeName;
        this.baseSalary = baseSalary;
    }

    void showEmployeeDetails() {
        System.out.println("Employee Name: " + employeeName);
        System.out.println("Base Salary: $" + baseSalary);
    }
}

class Manager extends Employee {
    double bonus = 10.0;

    Manager(String employeeName, double baseSalary) {
        super(employeeName, baseSalary);
    }

    void calculateTotalSalary() {
        double totalSalary = baseSalary + (baseSalary * bonus / 100);
        System.out.println("Total Salary with Bonus: $" + totalSalary);
    }
}

public class Singleinheritance2 {
    public static void main(String[] args) {
        Manager m1 = new Manager("Lohith", 6000);
```

```
        m1.showEmployeeDetails();
        m1.calculateTotalSalary();
    }
}
```

Output:

```
C:\Users\lohit\Downloads\Oops Report\single inheritance>javac Singleinheritance2.java

C:\Users\lohit\Downloads\Oops Report\single inheritance>java Singleinheritance2
Employee Name: Lohith
Base Salary: $6000.0
Total Salary with Bonus: $6600.0
```

5. Multilevel Inheritance

5.a) Company details

Code:

```
class Company {
   void companyInfo() {
      System.out.println("This is a tech company.");
   }
}

class Department extends Company {
   void departmentInfo() {
      System.out.println("Software Development Department.");
   }
}

class Employee extends Department {
   String name;

   Employee(String name) {
      this.name = name;
   }

   void work() {
      System.out.println(name + " is working on a project.");
   }

   void showName() {
      System.out.println("Employee Name: " + name);
   }
}

public class Multilevelinheritance1 {
   public static void main(String[] args) {
      Employee e1 = new Employee("Anjali");
      e1.companyInfo();
      e1.departmentInfo();
      e1.work();
      e1.showName();
   }
}
```

Output:

```
C:\Users\lohit\Downloads\Oops Report\multilevel inheritance>javac Multilevelinheritance1.java

C:\Users\lohit\Downloads\Oops Report\multilevel inheritance>java Multilevelinheritance1
This is a tech company.
Software Development Department.
Anjali is working on a project.
Employee Name: Anjali
```

5.b) University student grade

Code:

```
class University {
   void universityInfo() {
      System.out.println("Amrita University");
   }
}
class Course extends University {
   String co
urseName;

   Course(String course) {
      courseName = course;
   }

   void showCourseDetails() {
      System.out.println("Course: " + courseName);
   }
}

class Student extends Course {
   String studentName;
   double marks;

   Student(String course, String name, double mark) {
      super(course);
      studentName = name;
      marks = mark;
   }

   void calculateGrade() {
      String grade = (marks >= 90) ? "A+" : (marks >= 80) ? "A" : (marks >= 70) ? "B" : "C";
      System.out.println(studentName + " secured grade: " + grade);
   }
}
```

```
public class Multilevelinheritance2 {
   public static void main(String[] args) {
      Student s1 = new Student("Computer Science", "Lohith", 85);
      s1.universityInfo();
      s1.showCourseDetails();
      s1.calculateGrade();
   }
}
```

Output:

```
C:\Users\lohit\Downloads\Oops Report\multilevel inheritance>javac Multilevelinheritance2.java

C:\Users\lohit\Downloads\Oops Report\multilevel inheritance>java Multilevelinheritance2
Amrita University
Course: Computer Science
Lohith secured grade: A
```

6. Hierarchical Inheritance

6.a) Payment

Code:

```
class Payment {
   double amount;
   Payment(double amount) {
      this.amount = amount;
   }
   void showAmount() {
      System.out.println("Payment Amount: $" + amount);
   }
}

class CreditCard extends Payment {
   String cardNumber;
   CreditCard(double amount, String cardNumber) {
      super(amount);
      this.cardNumber = cardNumber;
   }
   void process() {
      System.out.println("Processing credit card ending with " +
        cardNumber.substring(cardNumber.length() - 4));
   }
}
class UPI extends Payment {
   String upiID;
   UPI(double amount, String upiID) {
      super(amount);
      this.upiID = upiID;
   }
   void payViaUPI() {
      System.out.println("Paying $" + amount + " via UPI ID: " + upiID);
   }
}
public class Hierarchicalinheritance1 {
   public static void main(String[] args) {
      CreditCard cc = new CreditCard(1200.50, "1234567890123456");
      UPI upi = new UPI(899.99, "lohith@upi");
      cc.showAmount();
      cc.process();
      upi.showAmount();
      upi.payViaUPI();
   }
}
```

Output:

```
C:\Users\lohit\Downloads\Oops Report\hierarchical inheritance>javac Hierarchicalinheritance1.java

C:\Users\lohit\Downloads\Oops Report\hierarchical inheritance>java Hierarchicalinheritance1
Payment Amount: $1200.5
Processing credit card ending with 3456
Payment Amount: $899.99
Paying $899.99 via UPI ID: lohith@upi
```

6.b) User log in

Code:

```
class User {
   String username;
   User(String username) {
      this.username = username;
   }
   void login() {
      System.out.println(username + " has logged in.");
   }
}

class Admin extends User {
   Admin(String username) {
      super(username);
   }
   void accessDashboard() {
      System.out.println("Admin " + username + " is accessing admin dashboard.");
   }
}

class Guest extends User {
   Guest(String username) {
      super(username);
   }
   void browseAsGuest() {
      System.out.println("Guest " + username + " is browsing the website.");
   }
}

public class Hierarchicalinheritance2 {
   public static void main(String[] args) {
      Admin admin = new Admin("admin123");
      Guest guest = new Guest("guest456");
      admin.login();
      admin.accessDashboard();
      System.out.println();
      guest.login();
      guest.browseAsGuest();
```

```
    }
}
```

Output:

```
C:\Users\lohit\Downloads\Oops Report\hierarchical inheritance>javac Hierarchicalinheritance2.java

C:\Users\lohit\Downloads\Oops Report\hierarchical inheritance>java Hierarchicalinheritance2
admin123 has logged in.
Admin admin123 is accessing admin dashboard.

guest456 has logged in.
Guest guest456 is browsing the website.
```

7. Hierarchical Inheritance

7.a) Vehicle

Code:

```
class Vehicle {
   String brand;
   Vehicle(String brand) {
      this.brand = brand;
   }
   void showBrand() {
      System.out.println("Vehicle Brand: " + brand);
   }
}

class Car extends Vehicle {
   int numberOfDoors;
   Car(String brand, int numberOfDoors) {
      super(brand);
      this.numberOfDoors = numberOfDoors;
   }
   void drive() {
      System.out.println(brand + " car is driving.");
   }
}

interface Chargeable {
   void chargeBattery();
}

class ElectricCar extends Car implements Chargeable {
   int batteryCapacity;
   ElectricCar(String brand, int numberOfDoors, int batteryCapacity) {
      super(brand, numberOfDoors);
      this.batteryCapacity = batteryCapacity;
   }
   public void chargeBattery() {
      System.out.println(brand + " electric car is charging with " + batteryCapacity +
" kWh battery.");
   }
}

public class Hybridinheritance1 {
   public static void main(String[] args) {
```

```
        ElectricCar e1 = new ElectricCar("Tesla", 4, 100);
        e1.showBrand();
        e1.drive();
         e1.chargeBattery();
    }
}
```

Output:

```
C:\Users\lohit\Downloads\Oops Report\hybrid inheritance>javac Hybridinheritance1.java

C:\Users\lohit\Downloads\Oops Report\hybrid inheritance>java Hybridinheritance1
Vehicle Brand: Tesla
Tesla car is driving.
Tesla electric car is charging with 100 kWh battery.
```

7.b) Gadget

Code:

```
class Gadget {
    String brand;
    Gadget(String brand) {
        this.brand = brand;
    }
    void info() {
        System.out.println("Brand: " + brand);
    }
}

class Phone extends Gadget {
    Phone(String brand) {
        super(brand);
    }
    void makeCall() {
        System.out.println(brand + " phone is making a call.");
    }
}

class Laptop extends Gadget {
    Laptop(String brand) {
        super(brand);
    }
    void compileCode() {
        System.out.println(brand + " laptop is compiling code.");
    }
```

```
}

interface SmartDevice {
   void connectToWiFi();
}

class SmartPhone extends Phone implements SmartDevice {
   SmartPhone(String brand) {
      super(brand);
   }
   public void connectToWiFi() {
      System.out.println(brand + " smartphone is connected to WiFi.");
   }
}

public class Hybridinheritance2 {
   public static void main(String[] args) {
      SmartPhone sp = new SmartPhone("Samsung");
      sp.info();
      sp.makeCall();
      sp.connectToWiFi();
   }
}
```

Output:

```
C:\Users\lohit\Downloads\Oops Report\hybrid inheritance>javac Hybridinheritance2.java

C:\Users\lohit\Downloads\Oops Report\hybrid inheritance>java Hybridinheritance2
Brand: Samsung
Samsung phone is making a call.
Samsung smartphone is connected to WiFi.
```

Polymorphism

8. Constructor programs

8.a) Movie

Code:

```
class Movie {
    String name;
    double rating;
    Movie(String name, double rating) {
        this.name = name;
        this.rating = rating;
    }

    Movie(Movie m) {
        this.name = m.name;
        this.rating = m.rating;
    }
    void display() {
        System.out.println("Movie: " + name + ", Rating: " + rating);
    }
}

public class Constructor1 {
    public static void main(String[] args) {
        Movie m1 = new Movie("Inception", 8.8);
        Movie m2 = new Movie("Interstellar", 8.6);
        m1.display();
        m2.display();
    }
}
```

Output:

```
C:\Users\lohit\Downloads\Oops Report\constructor>javac Constructor1.java

C:\Users\lohit\Downloads\Oops Report\constructor>java Constructor1
Movie: Inception, Rating: 8.8
Movie: Interstellar, Rating: 8.6
```

9. Constructor overloading programs

9.a) Book details

Code:

```
class Book {
    String title;
    String author;
    int pages;

    Book() {
        title = "Unknown Title";
        author = "Anonymous";
        pages = 100;
    }
    Book(String t, String a) {
        title = t;
        author = a;
        pages = 200;
    }
    Book(String t, String a, int p) {
        title = t;
        author = a;
        pages = p;
    }
    void displayBook() {
        System.out.println("Book: " + title + ", Author: " + author + ", Pages: " + pages);
    }
}

public class Constructoroverloading {
    public static void main(String[] args) {
        Book b1 = new Book();
        Book b2 = new Book("Java Basics", "James Gosling");
        Book b3 = new Book("Effective Java", "Joshua Bloch", 416);
        b1.displayBook();
        b2.displayBook();
        b3.displayBook();
    }
}
```

Output:

```
C:\Users\lohit\Downloads\Oops Report\constructor overloading>javac Constructoroverloading.java

C:\Users\lohit\Downloads\Oops Report\constructor overloading>java Constructoroverloading
Book: Unknown Title, Author: Anonymous, Pages: 100
Book: Java Basics, Author: James Gosling, Pages: 200
Book: Effective Java, Author: Joshua Bloch, Pages: 416
```

10. Method overloading programs

10.a) Calculate area

Code:

```
class AreaCalculator {
    void area(int side) {
        System.out.println("Area of square: " + (side * side));
    }

    void area(int length, int breadth) {
        System.out.println("Area of rectangle: " + (length * breadth));
    }

    void area(double base, double height) {
        System.out.println("Area of triangle: " + (0.5 * base * height));
    }
}

public class Methodoverloading1 {
    public static void main(String[] args) {
        AreaCalculator a = new AreaCalculator();
        a.area(5);
        a.area(4, 7);
        a.area(3.0, 6.0);
    }
}
```

Output:

```
C:\Users\lohit\Downloads\Oops Report\method overloading>javac Methodoverloading1.java

C:\Users\lohit\Downloads\Oops Report\method overloading>java Methodoverloading1
Area of square: 25
Area of rectangle: 28
Area of triangle: 9.0
```

10.b) Unit converter

Code:

```
class Converter {
    void convert(int km) {
        System.out.println(km + " km = " + (km * 1000) + " meters");
```

```
 }

    void convert(double kg) {
        System.out.println(kg + " kg = " + (kg * 2.20462) + " pounds");
    }

    void convert(int hours, int minutes) {
        int totalMinutes = (hours * 60) + minutes;
        System.out.println(hours + "h " + minutes + "m = " + totalMinutes + " minutes");
    }
}

public class Methodoverloading2 {
    public static void main(String[] args) {
        Converter c = new Converter();
        c.convert(5);
        c.convert(2.5);
        c.convert(1, 30);
    }
}
```

Output:

```
C:\Users\lohit\Downloads\Oops Report\method overloading>javac Methodoverloading2.java

C:\Users\lohit\Downloads\Oops Report\method overloading>java Methodoverloading2
5 km = 5000 meters
2.5 kg = 5.51155 pounds
1h 30m = 90 minutes
```

11. Method overiding programs

11.a) Calculate area

Code:

```
class Shape {
   double area() {
      return 0.0;
   }
}

class Circle extends Shape {
   double area() {
      return 3.14 * 5 * 5;
   }
}

class Square extends Shape {
   double area() {
      return 4 * 4;
   }
}

class Rectangle extends Shape {
   double area() {
      return 4 * 6;
   }
}

public class Methodoverriding1 {
   public static void main(String[] args) {
      Shape s;
      s = new Circle();
      System.out.println("Circle Area: " + s.area());
      s = new Square();
      System.out.println("Square Area: " + s.area());
      s = new Rectangle();
      System.out.println("Rectangle Area: " + s.area());
   }
}
```

Output:

```
C:\Users\lohit\Downloads\Oops Report\method overriding>javac Methodoverriding1.java

C:\Users\lohit\Downloads\Oops Report\method overriding>java Methodoverriding1
Circle Area: 78.5
Square Area: 16.0
Rectangle Area: 24.0
```

11.b) Languages

Code:

```
class Language {
    void sayHello() {
        System.out.println("Hello in a generic language");
    }
}

class English extends Language {
    void sayHello() {
        System.out.println("Hello");
    }
}

class Spanish extends Language {
    void sayHello() {
        System.out.println("Hola");
    }
}

class French extends Language {
    void sayHello() {
        System.out.println("Bonjour");
    }
}

public class Methodoverriding2 {
    public static void main(String[] args) {
        Language l;
        l = new English();
        l.sayHello();
        l = new Spanish();
        l.sayHello();
        l = new French();
        l.sayHello();
    }
}
```

Output:

```
C:\Users\lohit\Downloads\Oops Report\method overriding>javac Methodoverriding2.java

C:\Users\lohit\Downloads\Oops Report\method overriding>java Methodoverriding2
Hello
Hola
Bonjour
```

Abstraction

12. Interface programs

12.a) Math operations

Code:

```
interface MyMath {
    void Add(int a, int b);
    void Subtract(int a, int b);
    void Multiply(int a, int b);
    void Divide(int a, int b);
}

class Calculation implements MyMath {

    public void Add(int a, int b) {
        int result = a + b;
        System.out.println("Addition: " + result);
    }

    public void Subtract(int a, int b) {
        int result = a - b;
        System.out.println("Subtraction: " + result);
    }

    public void Multiply(int a, int b) {
        int result = a * b;
        System.out.println("Multiplication: " + result);
    }

    public void Divide(int a, int b) {
        if (b != 0) {
            int result = a / b;
            System.out.println("Division: " + result);
        } else {
            System.out.println("Cannot divide by zero.");
        }
    }
}

public class Interface1 {
    public static void main(String[] args) {
        MyMath num = new Calculation();
        num.Add(4, 6);
        num.Subtract(10, 3);
```

```
        num.Multiply(5, 2);
        num.Divide(9, 3);
        num.Divide(10, 0);
    }
}
```

Output:

```
C:\Users\lohit\Downloads\Oops Report\interface>javac Interface1.java

C:\Users\lohit\Downloads\Oops Report\interface>java Interface1
Addition: 10
Subtraction: 7
Multiplication: 10
Division: 3
Cannot divide by zero.
```

12.b) Temperature converter

Code:

```
interface Temperature {
    void ConvertToCelsius(int fahrenheit);
    void ConvertToFahrenheit(int celsius);
}

class Converter implements Temperature {
    public void ConvertToCelsius(int fahrenheit) {
        double celsius = (fahrenheit - 32) * 5 / 9;
        System.out.println(fahrenheit + " Fahrenheit is " + celsius + " Celsius");
    }

    public void ConvertToFahrenheit(int celsius) {
        double fahrenheit = (celsius * 9 / 5) + 32;
        System.out.println(celsius + " Celsius is " + fahrenheit + " Fahrenheit");
    }
}

class Interface2 {
    public static void main(String[] args) {
        Temperature temp = new Converter();
        temp.ConvertToCelsius(100);
        temp.ConvertToFahrenheit(37);
    }
}
```

Output:

```
C:\Users\lohit\Downloads\Oops Report\interface>javac Interface2.java

C:\Users\lohit\Downloads\Oops Report\interface>java Interface2
100 Fahrenheit is 37.0 Celsius
37 Celsius is 98.0 Fahrenheit
```

12.c) College

Code:

```
interface CollegeServices {
    void attendLecture(String lectureTopic);
    void submitAssignment(String assignmentName);
    void participateInEvent(String eventName);
    void enrollCourse(String courseName);

}

class CollegeStudent implements CollegeServices {
    private String studentName;

    public CollegeStudent(String studentName) {
        this.studentName = studentName;
    }

    public void enrollCourse(String courseName) {
        System.out.println(studentName + " has enrolled in " + courseName + ".");
    }

    public void attendLecture(String lectureTopic) {
        System.out.println(studentName + " is attending a lecture on " + lectureTopic +
".");
    }

    public void submitAssignment(String assignmentName) {
        System.out.println(studentName + " has submitted the " + assignmentName + "
assignment.");
    }

    public void participateInEvent(String eventName) {
        System.out.println(studentName + " is participating in the " + eventName + "
event.");
    }
}

public class Interface3 {
```

```
    public static void main(String[] args) {
        CollegeStudent student = new CollegeStudent("Lohith");

        student.enrollCourse("Java Programming");
        student.attendLecture("Object-Oriented Programming");
        student.submitAssignment("Java Basics");
        student.participateInEvent("College Fair");
    }
}
```

Output:

```
C:\Users\lohit\Downloads\Oops Report\interface>javac Interface3.java

C:\Users\lohit\Downloads\Oops Report\interface>java Interface3
Lohith has enrolled in Java Programming.
Lohith is attending a lecture on Object-Oriented Programming.
Lohith has submitted the Java Basics assignment.
Lohith is participating in the College Fair event.
```

12.d) String operations

Code:

```
import java.util.*;

interface StringOperations {
    void Concatenate(String str1, String str2);
    void Reverse(String str);
    void UpperCase(String str);
    void LowerCase(String str);
}

class StringManipulator implements StringOperations {
    public void Concatenate(String str1, String str2) {
        System.out.println("Concatenated: " + str1 + str2);
    }

    public void Reverse(String str) {
        String reversed = new StringBuilder(str).reverse().toString();
        System.out.println("Reversed: " + reversed);
    }

    public void UpperCase(String str) {
        System.out.println("Uppercase: " + str.toUpperCase());
    }
```

```
    public void LowerCase(String str) {
        System.out.println("Lowercase: " + str.toLowerCase());
    }
}

class Interface4 {
    public static void main(String[] args) {
        StringOperations strOps = new StringManipulator();
        strOps.Concatenate("Hello", "World");
        strOps.Reverse("Java");
        strOps.UpperCase("hello");
        strOps.LowerCase("WORLD");
    }
}
```

Output:

```
C:\Users\lohit\Downloads\Oops Report\interface>javac Interface4.java

C:\Users\lohit\Downloads\Oops Report\interface>java Interface4
Concatenated: HelloWorld
Reversed: avaJ
Uppercase: HELLO
Lowercase: world
```

13. Abstract class programs

13.a) Account

Code:

```
abstract class Account {
   double balance;

   Account(double balance) {
      this.balance = balance;
   }

   abstract void withdraw(double amount);
}

class SavingsAccount extends Account {
   SavingsAccount(double balance) {
      super(balance);
   }

   void withdraw(double amount) {
      if (balance >= amount) {
         balance -= amount;
         System.out.println("Withdrawn: " + amount + ", Remaining: " + balance);
      } else {
         System.out.println("Insufficient balance");
      }
   }
}

public class Abstractclass1 {
   public static void main(String[] args) {
      Account acc = new SavingsAccount(1000);
      acc.withdraw(400);
   }
}
```

Output:

```
C:\Users\lohit\Downloads\Oops Report\abstract class>javac Abstractclass1.java

C:\Users\lohit\Downloads\Oops Report\abstract class>java Abstractclass1
Withdrawn: 400.0, Remaining: 600.0
```

13.b) Media player

Code:

```
abstract class MediaPlayer {
   abstract void play();
   abstract void pause();
}

class AudioPlayer extends MediaPlayer {
   void play() {
      System.out.println("Playing audio...");
   }
   void pause() {
      System.out.println("Audio paused.");
   }
}

class VideoPlayer extends MediaPlayer {
   void play() {
      System.out.println("Playing video...");
   }
   void pause() {
      System.out.println("Video paused.");
   }
}

public class Abstractclass2 {
   public static void main(String[] args) {
      MediaPlayer player1 = new AudioPlayer();
      player1.play();
      player1.pause();

      MediaPlayer player2 = new VideoPlayer();
      player2.play();
      player2.pause();
   }
}
```

Output:

```
C:\Users\lohit\Downloads\Oops Report\abstract class>javac Abstractclass2.java

C:\Users\lohit\Downloads\Oops Report\abstract class>java Abstractclass2
Playing audio...
Audio paused.
Playing video...
Video paused.
```

13.c) Report details

Code:

```
abstract class Report {
    abstract void generate();
    abstract void export();
}

class SalesReport extends Report {
    void generate() {
        System.out.println("Generating Sales Report...");
    }
    void export() {
        System.out.println("Exporting Sales Report as PDF.");
    }
}

class AttendanceReport extends Report {
    void generate() {
        System.out.println("Generating Attendance Report...");
    }
    void export() {
        System.out.println("Exporting Attendance Report as Excel.");
    }
}

public class Abstractclass3 {
    public static void main(String[] args) {
        Report r1 = new SalesReport();
        r1.generate();
        r1.export();

        Report r2 = new AttendanceReport();
        r2.generate();
        r2.export();
    }
```

```
}
```

Output:

```
C:\Users\lohit\Downloads\Oops Report\abstract class>javac Abstractclass3.java

C:\Users\lohit\Downloads\Oops Report\abstract class>java Abstractclass3
Generating Sales Report...
Exporting Sales Report as PDF.
Generating Attendance Report...
Exporting Attendance Report as Excel.

C:\Users\lohit\Downloads\Oops Report\abstract class>
```

13.d) Editor

Code:

```
abstract class Editor {
    abstract void openFile();
    abstract void saveFile();
}

class TextEditor extends Editor {
    void openFile() {
        System.out.println("Opening text file...");
    }
    void saveFile() {
        System.out.println("Saving text file.");
    }
}

class CodeEditor extends Editor {
    void openFile() {
        System.out.println("Opening source code file...");
    }
    void saveFile() {
        System.out.println("Saving source code file.");
    }
}

public class Abstractclass4 {
    public static void main(String[] args) {
        Editor e1 = new TextEditor();
        e1.openFile();
        e1.saveFile();

        Editor e2 = new CodeEditor();
```

```
        e2.openFile();
        e2.saveFile();
    }
}
```

Output:

```
C:\Users\lohit\Downloads\Oops Report\abstract class>javac Abstractclass4.java

C:\Users\lohit\Downloads\Oops Report\abstract class>java Abstractclass4
Opening text file...
Saving text file.
Opening source code file...
Saving source code file.
```

Encapsulation

14. Encapsulation programs

14.a) Movie ticket

Code:

```
class MovieTicket {
    private String movieName;
    private int seats;
    private double pricePerSeat;

    public void setMovieName(String name) {
        movieName = name;
    }

    public void setSeats(int seats) {
        this.seats = seats > 0 ? seats : 0;
    }

    public void setPricePerSeat(double price) {
        this.pricePerSeat = price > 0 ? price : 0;
    }

    public double getTotalCost() {
        return seats * pricePerSeat;
    }

    public String getMovieName() {
        return movieName;
    }
}

public class Encapsulation1 {
    public static void main(String[] args) {
        MovieTicket ticket = new MovieTicket();
        ticket.setMovieName("Inception");
        ticket.setSeats(3);
        ticket.setPricePerSeat(12.5);
        System.out.println("Movie: " + ticket.getMovieName());
        System.out.println("Total Ticket Cost: $" + ticket.getTotalCost());
    }
}
```

Output:

```
C:\Users\lohit\Downloads\Oops Report\encapsulation>javac Encapsulation1.java

C:\Users\lohit\Downloads\Oops Report\encapsulation>java Encapsulation1
Movie: Inception
Total Ticket Cost: $37.5
```

14.b) Hotel booking

Code:

```
class HotelBooking {
    private int nights;
    private double pricePerNight;

    public void setNights(int nights) {
        this.nights = nights > 0 ? nights : 0;
    }

    public void setPricePerNight(double price) {
        this.pricePerNight = price > 0 ? price : 0;
    }

    public double getBookingCost() {
        return nights * pricePerNight;
    }
}

public class Encapsulation2 {
    public static void main(String[] args) {
        HotelBooking booking = new HotelBooking();
        booking.setNights(4);
        booking.setPricePerNight(100);
        System.out.println("Total Hotel Booking Cost: $" + booking.getBookingCost());
    }
}
```

Output:

```
C:\Users\lohit\Downloads\Oops Report\encapsulation>javac Encapsulation2.java

C:\Users\lohit\Downloads\Oops Report\encapsulation>java Encapsulation2
Total Hotel Booking Cost: $400.0
```

14.c) Loan interest

Code:

```
class Loan {
    private double principal;
    private double rate;
    private int years;

    public void setPrincipal(double principal) {
        this.principal = principal > 0 ? principal : 0;
    }

    public void setRate(double rate) {
        this.rate = rate > 0 ? rate : 0;
    }

    public void setYears(int years) {
        this.years = years > 0 ? years : 0;
    }

    public double getTotalInterest() {
        return (principal * rate * years) / 100;
    }
}

public class Encapsulation3 {
    public static void main(String[] args) {
        Loan loan = new Loan();
        loan.setPrincipal(10000);
        loan.setRate(5);
        loan.setYears(3);
        System.out.println("Total Interest: $" + loan.getTotalInterest());
    }
}
```

Output:

```
C:\Users\lohit\Downloads\Oops Report\encapsulation>javac Encapsulation3.java

C:\Users\lohit\Downloads\Oops Report\encapsulation>java Encapsulation3
Total Interest: $1500.0
```

14.d) Rectangle

Code:

```
class Rectangle {
    private double length;
    private double width;

    public void setLength(double length) {
        this.length = length > 0 ? length : 0;
    }

    public void setWidth(double width) {
        this.width = width > 0 ? width : 0;
    }

    public double getArea() {
        return length * width;
    }

    public double getPerimeter() {
        return 2 * (length + width);
    }
}

public class Encapsulation4 {
    public static void main(String[] args) {
        Rectangle rectangle = new Rectangle();
        rectangle.setLength(5);
        rectangle.setWidth(3);
        System.out.println("Area: " + rectangle.getArea());
        System.out.println("Perimeter: " + rectangle.getPerimeter());
    }
}
```

Output:

```
C:\Users\lohit\Downloads\Oops Report\encapsulation>javac Encapsulation4.java

C:\Users\lohit\Downloads\Oops Report\encapsulation>java Encapsulation4
Area: 15.0
Perimeter: 16.0
```

15. Package programs

15.a) Exponents

Code:

```
package power;

public class exponents {
    public int square(int x) {
        return x * x;
    }

    public int cube(int x) {
        return x * x * x;
    }
}


import power.exponents;

public class Package1 {
    public static void main(String[] args) {
        exponents num = new exponents();
        System.out.println("Square of 3: " + num.square(3));
        System.out.println("Cube of 3: " + num.cube(3));
    }
}
```

Output:

```
C:\Users\lohit\Downloads\Oops Report\package>javac -d . exponents.java

C:\Users\lohit\Downloads\Oops Report\package>javac Package1.java

C:\Users\lohit\Downloads\Oops Report\package>java Package1
Square of 3: 9
Cube of 3: 27
```

15.b) Temperature converter

Code:

```
package converter;

public class TemperatureConverter {
    public double toFahrenheit(double celsius) {
        return (celsius * 9/5) + 32;
    }

    public double toCelsius(double fahrenheit) {
        return (fahrenheit - 32) * 5/9;
    }
}

import converter.TemperatureConverter;

public class Package2 {
    public static void main(String[] args) {
        TemperatureConverter tc = new TemperatureConverter();
        System.out.println("25°C in Fahrenheit: " + tc.toFahrenheit(25));
        System.out.println("77°F in Celsius: " + tc.toCelsius(77));
    }
}
```

Output:

```
C:\Users\lohit\Downloads\Oops Report\package>javac -d . TemperatureConverter.java

C:\Users\lohit\Downloads\Oops Report\package>javac Package2.java

C:\Users\lohit\Downloads\Oops Report\package>java Package2
25°C in Fahrenheit: 77.0
77°F in Celsius: 25.0
```

15.c) Sort even numbers

Code:

```
import java.util.List;
import java.util.stream.Collectors;
import java.util.function.Predicate;

public class Package3 {
    public static void main(String[] args) {
        List<Integer> nums = List.of(10, 25, 30, 45, 60);
```

```
 Predicate<Integer> isEven = x -> x % 2 == 0;

        List<Integer> evenNums = nums.stream().filter(isEven).collect(Collectors.toList());
        System.out.println("Even numbers: " + evenNums);
    }
}
```

Output:

```
C:\Users\lohit\Downloads\Oops Report\package>javac Package3.java

C:\Users\lohit\Downloads\Oops Report\package>java Package3
Even numbers: [10, 30, 60]
```

15.d) Date

Code:

```
import java.util.Date;
import java.text.DateFormat;
import java.lang.System;

public class Package4 {
    public static void main(String[] args) {
        Date now = new Date();
        DateFormat df = DateFormat.getDateTimeInstance(DateFormat.FULL,
DateFormat.FULL);
        System.out.println("Formatted Date: " + df.format(now));
    }
}
```

Output:

```
C:\Users\lohit\Downloads\Oops Report\package>javac Package4.java

C:\Users\lohit\Downloads\Oops Report\package>java Package4
Formatted Date: Friday 4 April, 2025, 11:17:40?pm India Standard Time
```

16. Excetion handling programs

16.a) Check input format

Code:

```
import java.util.Scanner;

public class Exceptionhandling1 {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        System.out.print("Enter a value: ");
        String input = sc.nextLine();

        try {
            double number = Double.parseDouble(input);
            System.out.println("Valid number format: " + number);
        } catch (NumberFormatException e) {
            System.out.println("Invalid number format!");
        }
    }
}
```

Output:

```
C:\Users\lohit\Downloads\Oops Report\exception handling>javac Exceptionhandling1.java

C:\Users\lohit\Downloads\Oops Report\exception handling>java Exceptionhandling1
Enter a value: 3
Valid number format: 3.0

C:\Users\lohit\Downloads\Oops Report\exception handling>java Exceptionhandling1
Enter a value: ab
Invalid number format!
```

16.b) Password check

Code:

```
import java.util.Scanner;

class ShortPasswordException extends Exception {
    public ShortPasswordException(String message) {
        super(message);
    }
}

public class Exceptionhandling2 {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter your password: ");
        String password = scanner.nextLine();

        try {
            if (password.length() < 6) {
                throw new ShortPasswordException("Password must be at least 6 characters");
            }
            System.out.println("Password accepted");
        } catch (ShortPasswordException e) {
            System.out.println("Weak password: " + e.getMessage());
        }

        scanner.close();
    }
}
```

Output:

```
C:\Users\lohit\Downloads\Oops Report\exception handling>javac Exceptionhandling2.java

C:\Users\lohit\Downloads\Oops Report\exception handling>java Exceptionhandling2
Enter your password: 3456
Weak password: Password must be at least 6 characters
```

16.c) Email validation

Code:

```
import java.util.Scanner;

class InvalidEmailException extends Exception {
    public InvalidEmailException(String message) {
        super(message);
    }
}

public class Exceptionhandling3 {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter your email: ");
        String email = sc.nextLine();

        try {
            if (!email.contains("@") || !email.contains(".")) {
                throw new InvalidEmailException("Email must contain '@' and '.'");
            }
            System.out.println("Email is valid.");
        } catch (InvalidEmailException e) {
            System.out.println("Invalid email: " + e.getMessage());
        }

        sc.close();
    }
}
```

Output:

```
C:\Users\lohit\Downloads\Oops Report\exception handling>javac Exceptionhandling3.java

C:\Users\lohit\Downloads\Oops Report\exception handling>java Exceptionhandling3
Enter your email: lohith
Invalid email: Email must contain '@' and '.'

C:\Users\lohit\Downloads\Oops Report\exception handling>java Exceptionhandling3
Enter your email: lohith@gmail.com
Email is valid.
```

16.d) Age check

Code:

```
import java.util.Scanner;

class UnderageException extends Exception {
    public UnderageException(String message) {
        super(message);
    }
}

public class Exceptionhandling4 {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter your age: ");
        int age = sc.nextInt();

        try {
            if (age < 18) {
                throw new UnderageException("You must be at least 18 years old to vote.");
            }
            System.out.println("You are eligible to vote.");
        } catch (UnderageException e) {
            System.out.println("Validation Error: " + e.getMessage());
        }

        sc.close();
    }
}
```

Output:

```
C:\Users\lohit\Downloads\Oops Report\exception handling>javac Exceptionhandling4.java

C:\Users\lohit\Downloads\Oops Report\exception handling>java Exceptionhandling4
Enter your age: 16
Validation Error: You must be at least 18 years old to vote.

C:\Users\lohit\Downloads\Oops Report\exception handling>java Exceptionhandling4
Enter your age: 20
You are eligible to vote.
```

17. File handling programs

17.a) Deleting file

Code:

```
import java.io.File;
public class Filehandling1 {
  public static void main(String args[]) {
    final String fileName = "file3.txt";
    File objFile = new File(fileName);

    if (objFile.exists() == true) {
      if (objFile.delete()) {
        System.out.println(objFile.getName() + " deleted successfully.");
      } else {
        System.out.println("File deletion failed!!!");
      }
    } else {
      System.out.println("File does not exist!!!");
    }
  }
}
```

Output:

```
C:\Users\lohit\Downloads\Oops Report\file handling>javac Filehandling1.java

C:\Users\lohit\Downloads\Oops Report\file handling>java Filehandling1
file3.txt deleted successfully.

C:\Users\lohit\Downloads\Oops Report\file handling>java Filehandling1
File does not exist!!!
```

17.b) Get file details

Code:

```
import java.io.File;

public class Filehandling2 {
  public static void main(String[] args) {
```

```
    File file = new File("filee.txt");

    if (file.exists()) {
      System.out.println("File name: " + file.getName());
      System.out.println("Path: " + file.getAbsolutePath());
      System.out.println("Writable: " + file.canWrite());
      System.out.println("Readable: " + file.canRead());
      System.out.println("File size in bytes: " + file.length());
    } else {
      System.out.println("File not found.");
    }
  }
}
```

Output:

```
C:\Users\lohit\Downloads\Oops Report\file handling>javac Filehandling2.java

C:\Users\lohit\Downloads\Oops Report\file handling>java Filehandling2
File name: filee.txt
Path: C:\Users\lohit\Downloads\Oops Report\file handling\filee.txt
Writable: true
Readable: true
File size in bytes: 0
```

17.c) Create new file

Code:

```
import java.io.File;
import java.io.IOException;

public class Filehandling3 {
  public static void main(String[] args) {
    try {
      File file = new File("newfile.txt");
      if (file.createNewFile()) {
        System.out.println("File created: " + file.getName());
      } else {
        System.out.println("File already exists.");
      }
    } catch (IOException e) {
      System.out.println("An error occurred.");
      e.printStackTrace();
    }
  }
}
```

Output:

```
C:\Users\lohit\Downloads\Oops Report\file handling>javac Filehandling3.java

C:\Users\lohit\Downloads\Oops Report\file handling>java Filehandling3
File created: newfile.txt

C:\Users\lohit\Downloads\Oops Report\file handling>java Filehandling3
File already exists.
```

17.d) Write in a new file

Code:

```
import java.io.BufferedWriter;
import java.io.FileWriter;

public class Filehandling4 {
    public static void main(String[] args) {
        try{
            FileWriter file=new FileWriter("output.txt",true);
            BufferedWriter b=new BufferedWriter(file);
            b.write("it creates a new a file");
            b.newLine();
            b.write("and write the contents in it");
            b.close();
        }
        catch(Exception e){
            System.out.println("error");
    }
    }
}
```

Output:

```
C:\Users\lohit\Downloads\Oops Report\file handling>javac Filehandling4.java

C:\Users\lohit\Downloads\Oops Report\file handling>java Filehandling4

C:\Users\lohit\Downloads\Oops Report\file handling>
```