LAB RECORD

23CSE111- Object Oriented Programming

Submitted by

CH.SC.U4CSE24015 – S DEEPAK BHARATHWAJ

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE AND ENGINEERING

AMRITA VISHWA VIDYAPEETHAM

AMRITA SCHOOL OF COMPUTING

CHENNAI

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.U4CSE24015 – S DEEPAK BHARATHWAJ*** 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 / /2025

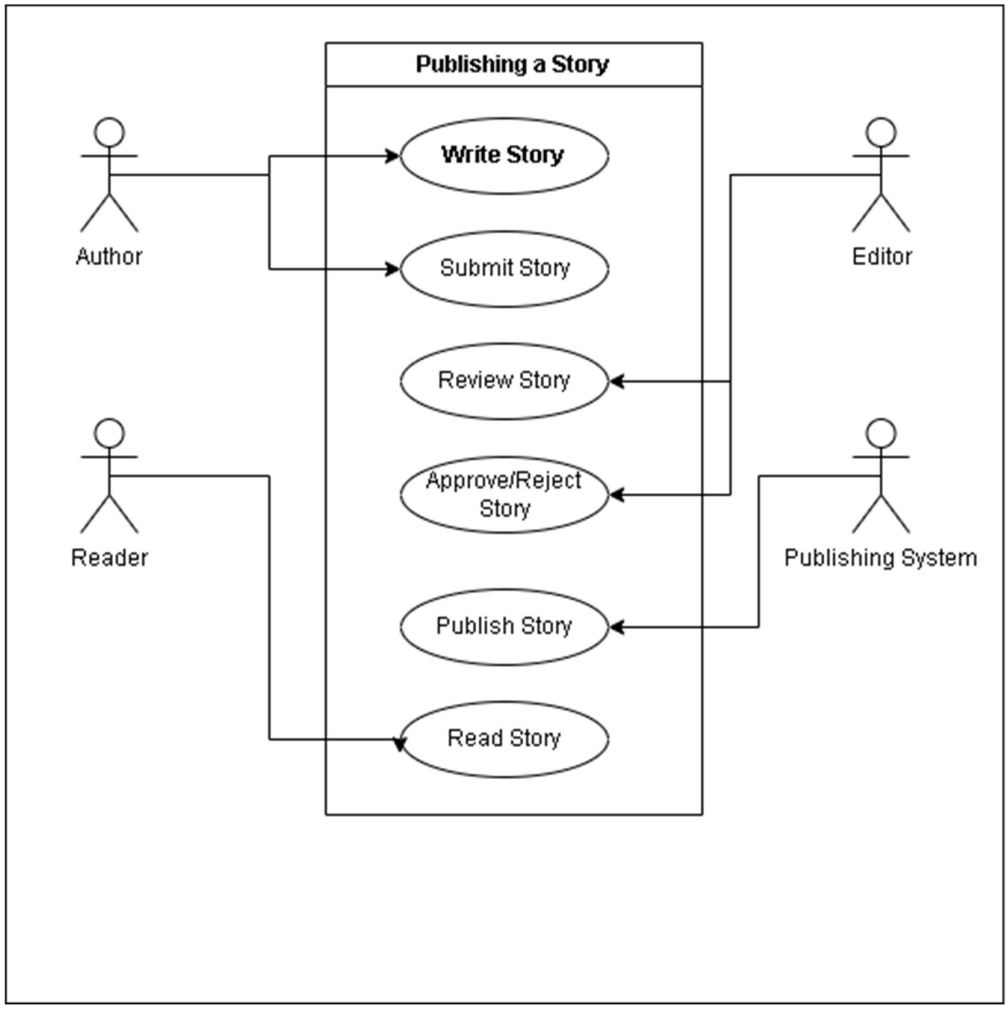
Internal Examiner 1 Internal Examiner 2

INDEX

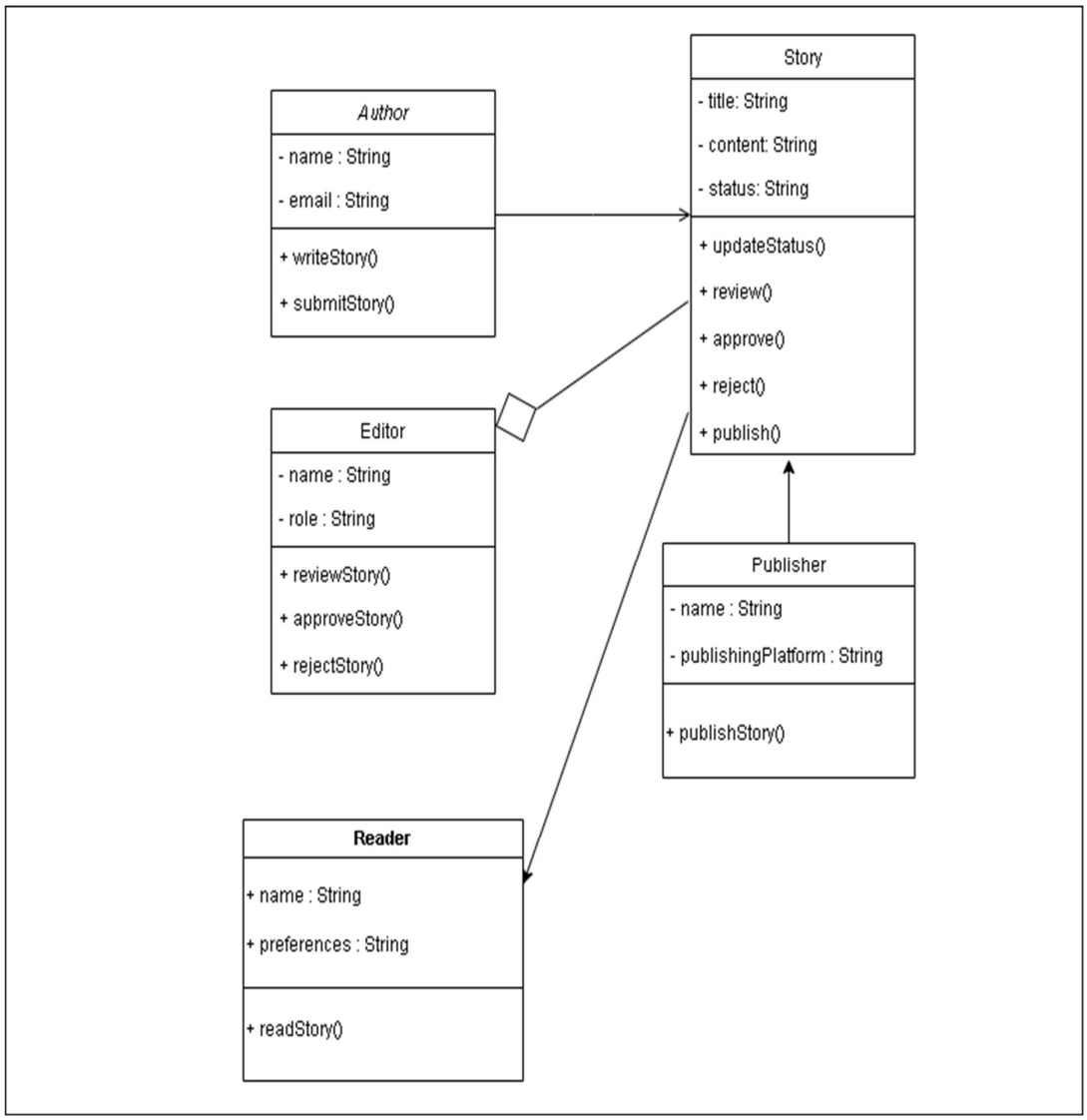
UML Diagrams

1) Publish a Story

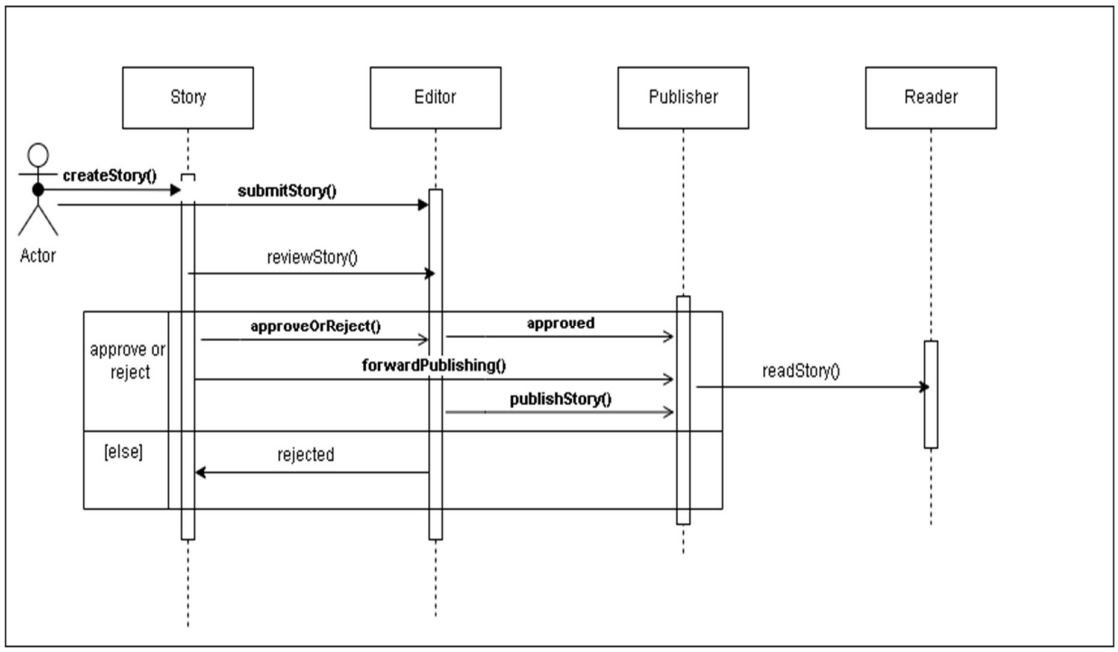
a) Use Case :



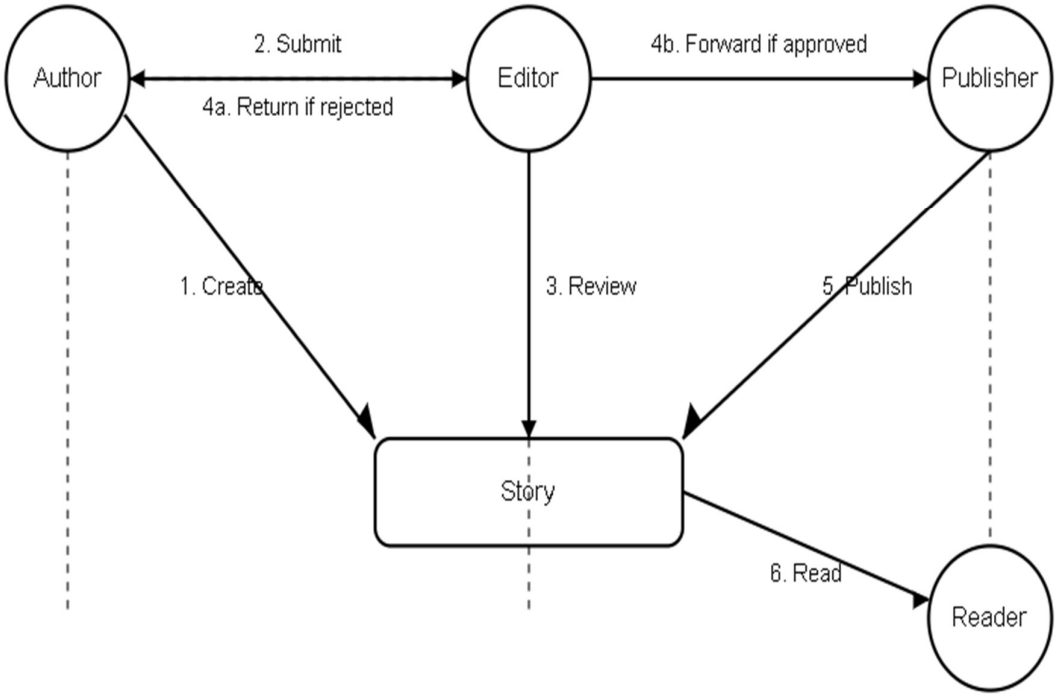
b) Class :


Author
- name : String
- email : String
+ writeStory()
+ submitStory()
Story
- title: String
- content: String
- status: String
+ updateStatus()
+ review()
+ approve()
+ reject()
+ publish()
Editor
- name : String
- role : String
+ reviewStory()
+ approveStory()
+ rejectStory()
Publisher
- name : String
- publishingPlatform : String
+ publishStory()
Reader
+ name : String
+ preferences : String
+ readStory()

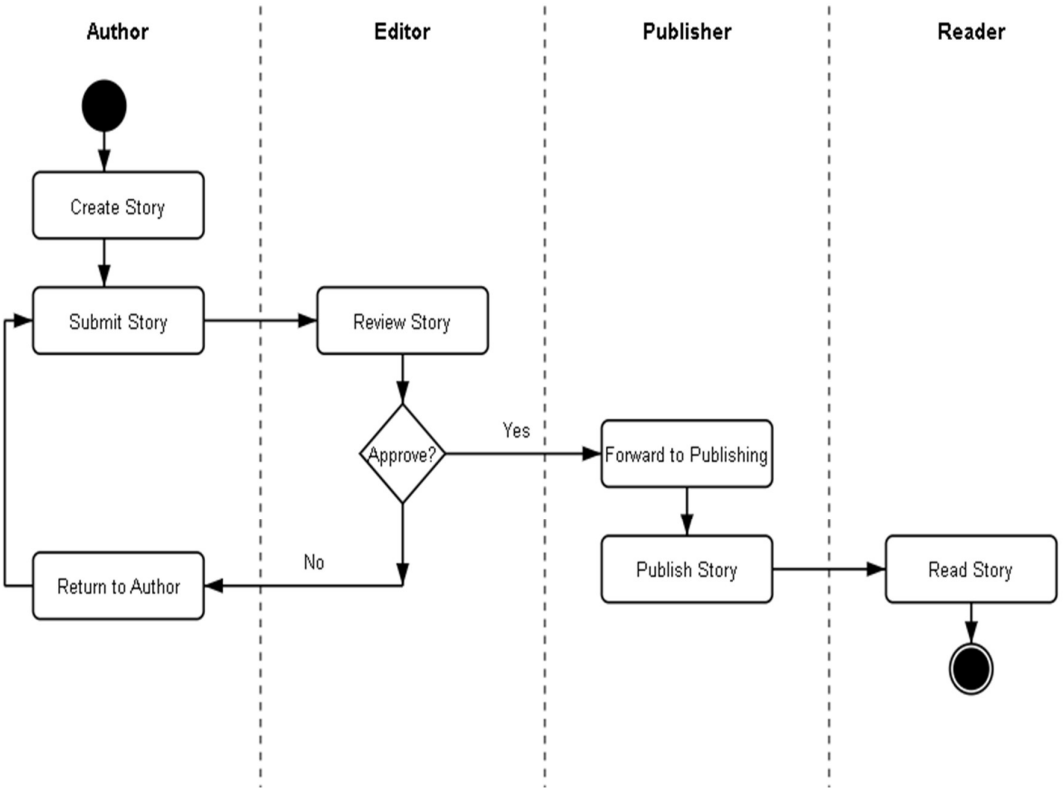
c) Sequence



d) Collaboration :

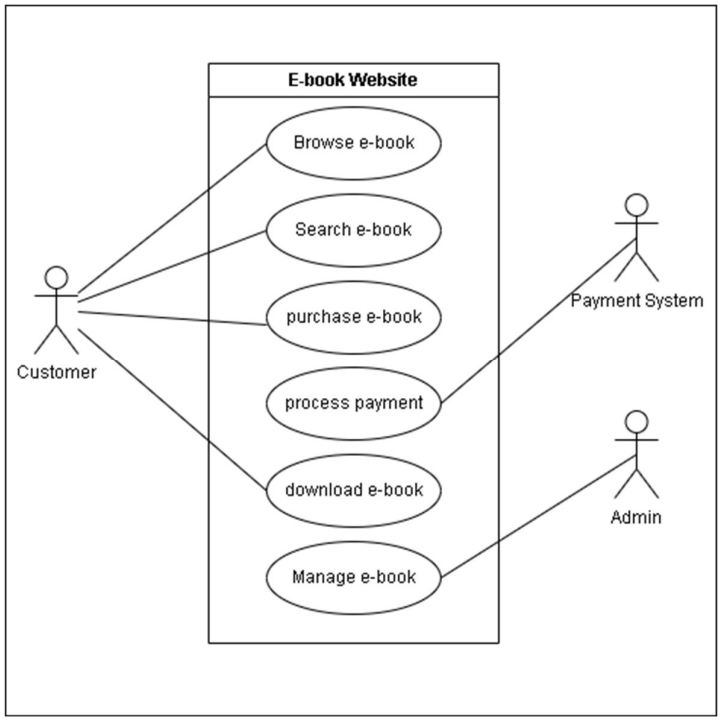


e) Activity :

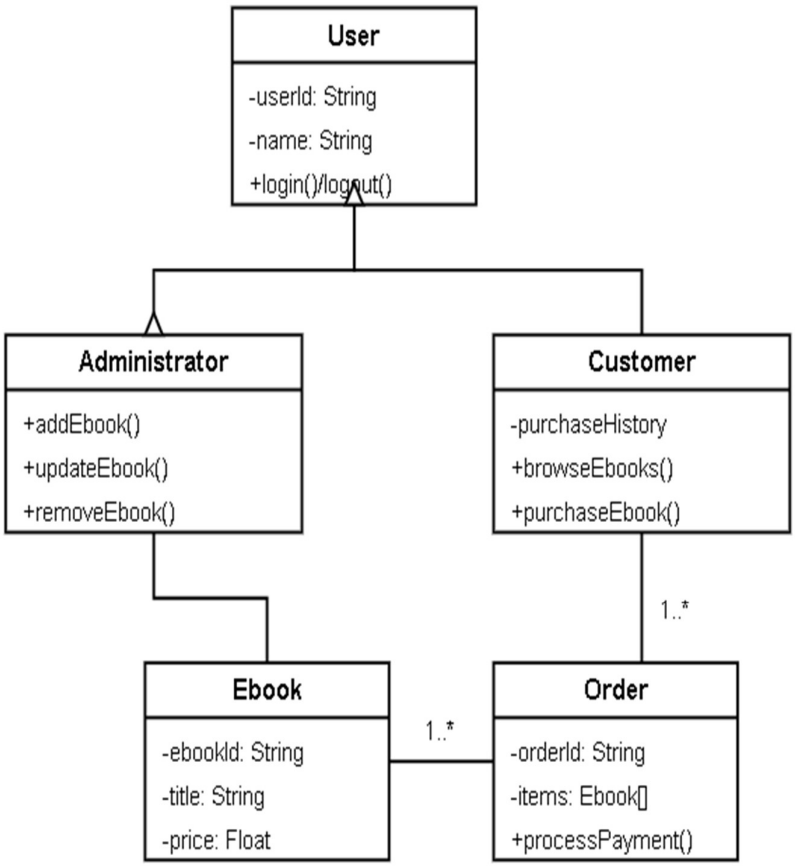


2) Online BookStore

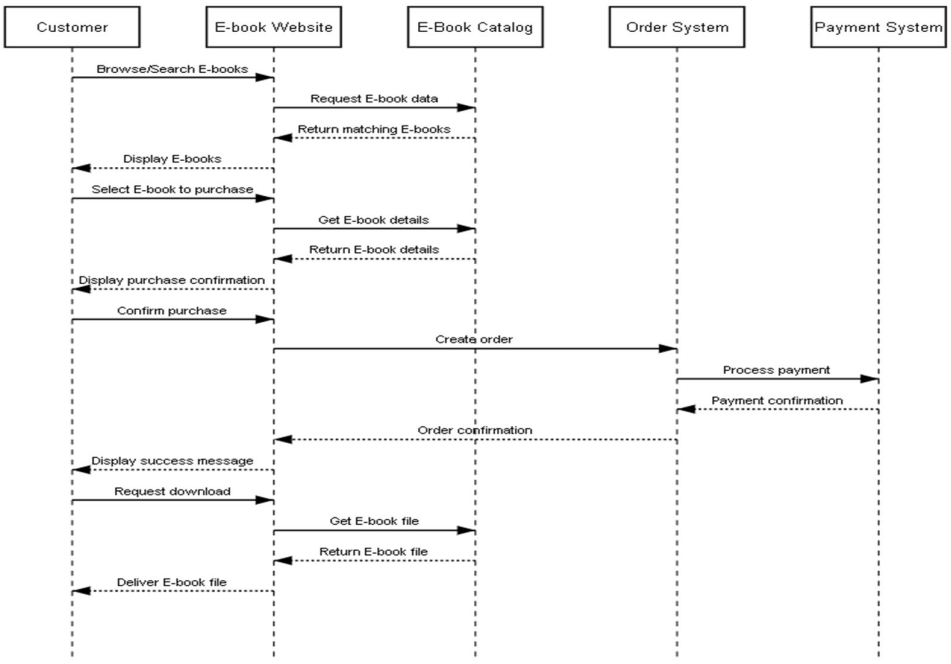
a) Use Case :



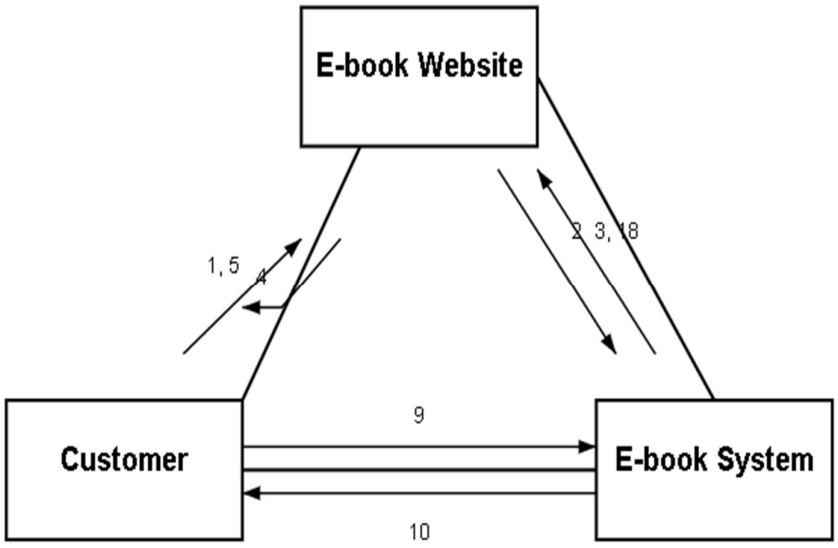
b) Class :



c) Sequence :



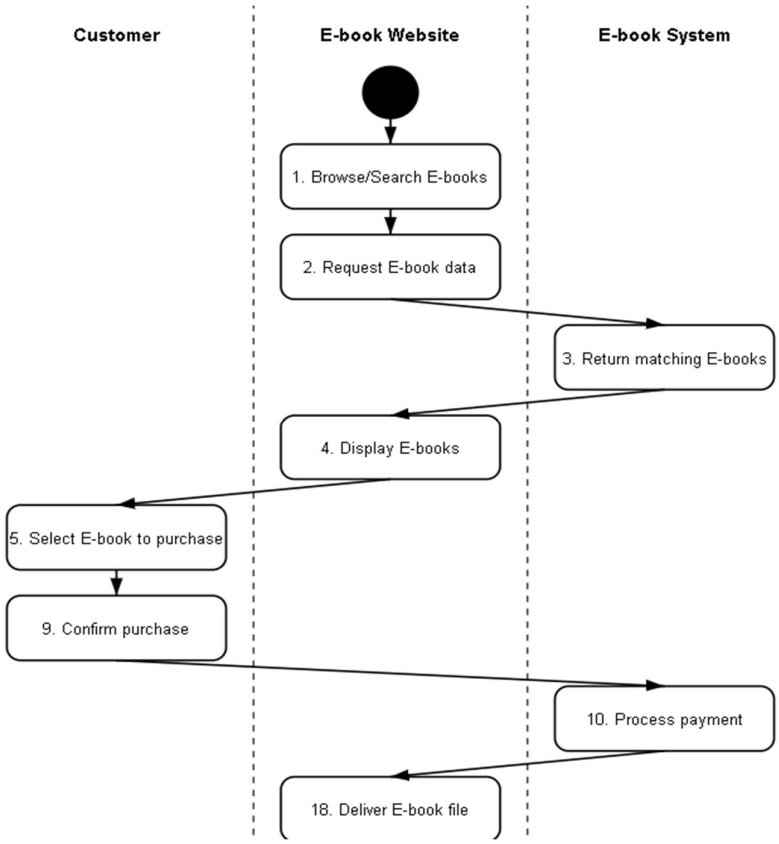
d) Collaboration :



Message Key:

1: Browse/Search E-books
2: Request E-book data
3: Return matching E-books
4: Display E-books
5: Select E-book to purchase
9: Confirm purchase
10: Process payment
18: Deliver E-book file

e) Activity :



3. Basic Java Programs

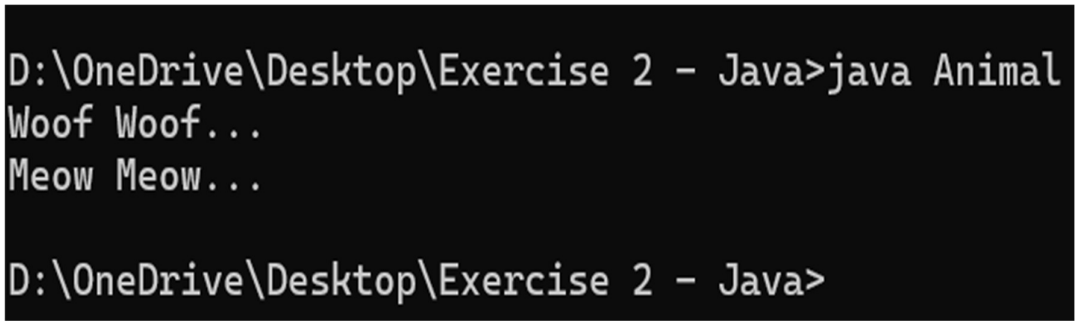
3a) Animal

CODE :

```
class dog extends Animal{
   @Override
   void make_sound(){
      System.out.println("Woof Woof...");
}
}

class cat extends Animal{
   @Override
   void make_sound(){
      System.out.println("Meow Meow...");
   }
```

```
}

public class Animal {

   void make_sound(){
      System.out.println("Sound...");
   }

   public static void main(String[] args) {
      Animal woof = new dog();
      Animal meow = new cat();
      woof.make_sound();
      meow.make_sound();
   }
    }
```

OUTPUT:



3b) Building

CODE :

```
class apartment extends Building{
   int no_of_apartments;
}

class house extends Building{
   int house;
}

public class Building {
    int floor;
```

```
  public static void main(String[] args) {
    house hou = new house();
    Building h = new house();
    apartment a = new apartment();
    System.out.println("number of floors in this floor is : "+(h.floor=10));
    System.out.println("number of apartments is : "+(a.no_of_apartments=10));
    System.out.println("number of houses in these apartments is : "+(hou.house=100));
  }
   }
```

OUTPUT:

```
D:\OneDrive\Desktop\Exercise 2 - Java>java Building
number of floors in this floor is : 10
number of apartments is : 10
number of houses in these apartments is : 100

D:\OneDrive\Desktop\Exercise 2 - Java>
```

3c) Palindrome

CODE :

```
    import java.util.Scanner;
class palindrome {
 public static void main(String[] args) {
  Scanner obj = new Scanner(System.in);
  System.out.println("Enter the number :");
  int num = obj.nextInt();

  int temp = num;
  int rev=0;

  while(num>0) {
  int dig=num%10;
  rev=rev*10+dig;
  num=num/10;
  }

  if(temp == rev) {
```

```
   System.out.println("The number "+rev+" is a palindrome!");
   }
   else{
   System.out.println("The number "+rev+" isn't a palindrome!");
   }
 }
   }
```

OUTPUT:

```
D:\OneDrive\Desktop\Exercise 2 - Java>java palindrome
Enter the number :
646
The number 646 is a palindrome!

D:\OneDrive\Desktop\Exercise 2 - Java>
```

3d) Vehicle

CODE :

```
class Vehicle{
   void display(){
     System.out.println("Vehicle goes Vroom...");
   }
   void engine(){ // it can be modified
   }
}

class bike extends Vehicle{
   @Override
   void display(){
     System.out.println("Bike goes Vroom...");
   }
   @Override
   void engine(){ // im modifying it
     System.out.println("mehhhhh");
   }
}

class car extends Vehicle{
```

```
    @Override
    void display(){
        System.out.println("Car go meow...");
    }
}
public class Poly {

    public static void main(String[] args) {
        Vehicle v = new Vehicle();
        Vehicle b = new bike();
        Vehicle c = new car();
        b.display();
        c.display();
        v.display();
        b.engine();
    }
     }
```

OUTPUT:

```
D:\OneDrive\Desktop\Exercise 2 - Java>java Poly
Bike goes Vroom...
Car go meow...
Vehicle goes Vroom...
mehhhhh

D:\OneDrive\Desktop\Exercise 2 - Java>
```

3e) area

CODE :

```
    import java.util.Scanner;
public class area
{
  public static void main (String args[])
   {
     Scanner obj = new Scanner(System.in);
     Scanner obj2 = new Scanner(System.in);
```

```
    System.out.println("Enter the base : ");
    int b = obj.nextInt();
    System.out.println("Enter the height : ");
    int h = obj2.nextInt();
     int area = ( b*h) / 2 ;
     System.out.println("Area of Triangle is: "+area);
      }}
```

OUTPUT:

```
D:\OneDrive\Desktop\Exercise 2 - Java>java area
Enter the base :
5
Enter the height :
2
Area of Triangle is: 5

D:\OneDrive\Desktop\Exercise 2 - Java>
```

3f) Armstrong

CODE :

```
import java.util.Scanner;
class armstrong {
  public static void main(String[] args) {
    Scanner obj = new Scanner(System.in);
    System.out.println("Enter the number :");
    int num = obj.nextInt();

    int temp = num;
    int sum=0;

    while(temp>0) {
    int dig=temp%10;
    sum+=Math.pow(dig, 3);
    temp=temp/10;
    }
```

```
  if(num == sum) {
  System.out.println("The number "+num+" is a Armstrong number!");
  }
  else{
  System.out.println("The number "+num+" isn't a Armstrong number!");
  }
 }
  }
```

OUTPUT:

```
D:\OneDrive\Desktop\Exercise 2 - Java>javac armstrong.java

D:\OneDrive\Desktop\Exercise 2 - Java>java armstrong
Enter the number :
153
The number 153 is a Armstrong number!

D:\OneDrive\Desktop\Exercise 2 - Java>
```

3g) Fibonacci

CODE :

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

  int n = 10, firstTerm = 0, secondTerm = 1;
  System.out.println("Fibonacci Series till " + n + " terms:");

  for (int i = 1; i <= n; ++i) {
   System.out.print(firstTerm + ", ");

   // compute the next term
   int nextTerm = firstTerm + secondTerm;
   firstTerm = secondTerm;
   secondTerm = nextTerm;
  }
 }
  }
```

OUTPUT:

```
D:\OneDrive\Desktop\Exercise 2 - Java>javac fibonacci.java

D:\OneDrive\Desktop\Exercise 2 - Java>java fibonacci
Fibonacci Series till 10 terms:
0, 1, 1, 2, 3, 5, 8, 13, 21, 34,
D:\OneDrive\Desktop\Exercise 2 - Java>
```

3h) Prime

CODE :

```
import java.util.Scanner;
public class prime{

 public static void main(String[] args) {

  Scanner obj = new Scanner(System.in);
  System.out.println("Enter the number : ");
  int num = obj.nextInt();

  int i = 2;
  boolean flag = false;

  if (num == 0 || num == 1) {
    flag = true;
  }

  while (i <= num / 2) {

   if (num % i == 0) {
    flag = true;
    break;
   }

   ++i;
  }

  if (!flag)
```

```
    System.out.println(num + " is a prime number.");
   else
    System.out.println(num + " is not a prime number.");
 }
   }
```

OUTPUT:

```
D:\OneDrive\Desktop\Exercise 2 - Java>java prime
Enter the number :
97
97 is a prime number.

D:\OneDrive\Desktop\Exercise 2 - Java>
```

3i)Factorial

CODE :

```
    import java.util.Scanner;
class factorial{
 public static void main(String args[]){
  Scanner obj = new Scanner(System.in);
  int fact=1;
  System.out.println("Enter the number : ");
  int number = obj.nextInt();

  for(int i=1;i<=number;i++){
    fact=fact*i;
  }
  System.out.println("Factorial of "+number+" is: "+fact);
 }
   }
```

OUTPUT:

```
D:\OneDrive\Desktop\Exercise 2 - Java>java factorial
Enter the number :
8
Factorial of 8 is: 40320

D:\OneDrive\Desktop\Exercise 2 - Java>Z
```

3j)Hospital

CODE :

```
 class Person{
String name;
int age;

Person(String name, int age){
   this.name = name;
   this.age = age;
}

void display(){
   System.out.println("Name is : "+name);
   System.out.println("Age is : "+age);
}
}

class Doctor extends Person{
String special;

Doctor(String name,int age,String special){
   super(name,age);
   this.special = special;
}

@Override
void display(){
   super.display();
   System.out.println("Specialazation is : "+special);
}
```

```
}

public class Hospital {

   public static void main(String[] args) {
      Doctor doc = new Doctor("Deepak",18,"Neurologists");
      doc.display();
   }
    }
```

OUTPUT:

```
D:\OneDrive\Desktop\Exercise 2 - Java>java Hospital
Name is : Deepak
Age is : 18
Specialazation is : Neurologists

D:\OneDrive\Desktop\Exercise 2 - Java>
```

4A: SINGLE INHERITANCE 1

CODE:

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

   void setBook(String title, int pages) {
      this.title = title;
      this.pages = pages;
   }
}

class EBook extends Book {
   double fileSizeMB;

   void setFileSize(double size) {
      fileSizeMB = size;
   }

   void downloadTime(double speedMbps) {
      double time = fileSizeMB / speedMbps;
      System.out.println("Downloading \"" + title + "\"...");
      System.out.println("Estimated time: " + String.format("%.2f", time) + " seconds");
   }
}

public class Single1 {
   public static void main(String[] args) {
      EBook ebook = new EBook();
      ebook.setBook("Java for Beginners", 320);
      ebook.setFileSize(12.5);
      ebook.downloadTime(5);
   }
}
```

OUTPUT:

```
D:\OneDrive\Desktop\OOPS\assignment\Inheritance>java Single1
Downloading "Java for Beginners"...
Estimated time: 2.50 seconds

D:\OneDrive\Desktop\OOPS\assignment\Inheritance>
```

4B:SINGLE INHERITANCE 2

CODE:

```
class Game {
   String name;
   int players;

   void setGame(String name, int players) {
      this.name = name;
      this.players = players;
   }
}

class OnlineGame extends Game {
   int ping;
   void setPing(int ping) {
      this.ping = ping;
   }

   void showPerformance() {
      System.out.println("Game: " + name);
      System.out.println("Players Online: " + players);
      if (ping < 100)
         System.out.println("Ping: " + ping + "ms (Good)");
      else
         System.out.println("Ping: " + ping + "ms (Laggy)");
   }
}

public class Single2 {
   public static void main(String[] args) {
      OnlineGame og = new OnlineGame();
      og.setGame("Valorant", 9000);
      og.setPing(85);
      og.showPerformance();
   }
}
```

OUTPUT:

```
D:\OneDrive\Desktop\OOPS\assignment\Inheritance>java Single2
Game: Valorant
Players Online: 9000
Ping: 85ms (Good)
```

5A: MULTILEVEL INHERITANCE 1

CODE:

```
class Vehicle {
   int speed;
   int fuel;

   Vehicle(int speed, int fuel) {
      this.speed = speed;
      this.fuel = fuel;
   }

   void showStatus() {
      System.out.println("Speed: " + speed + " km/h");
      System.out.println("Fuel Level: " + fuel + "%");
   }

   void accelerate() {
      speed += 10;
      fuel -= 5;
      System.out.println("Vehicle accelerated.");
   }
}

class Car extends Vehicle {
   boolean airConditioningOn;

   Car(int speed, int fuel) {
      super(speed, fuel);
      this.airConditioningOn = false;
   }

   void toggleAC() {
      airConditioningOn = !airConditioningOn;
      System.out.println("AC turned " + (airConditioningOn ? "ON" : "OFF"));
      if (airConditioningOn) fuel -= 2;
   }

   @Override
   void showStatus() {
      super.showStatus();
      System.out.println("Air Conditioning: " + (airConditioningOn ? "ON" : "OFF"));
   }
```

```
}

class SmartCar extends Car {
  String gpsLocation;

  SmartCar(int speed, int fuel, String gpsLocation) {
    super(speed, fuel);
    this.gpsLocation = gpsLocation;
  }

  void updateGPS(String newLocation) {
    gpsLocation = newLocation;
    System.out.println("GPS updated to: " + gpsLocation);
  }

  void showStatus() {
    super.showStatus();
    System.out.println("GPS Location: " + gpsLocation);
  }
}

public class Multi1 {
  public static void main(String[] args) {
    SmartCar myCar = new SmartCar(60, 80, "Downtown");

    System.out.println("--- Initial Status ---");
    myCar.showStatus();

    System.out.println("\n--- After Actions ---");
    myCar.accelerate();
    myCar.toggleAC();
    myCar.updateGPS("Uptown");

    System.out.println("\n--- Updated Status ---");
    myCar.showStatus();
  }
}
```

OUTPUT:

```
D:\OneDrive\Desktop\OOPS\assignment\Multi1>java Multi1
--- Initial Status ---
Speed: 60 km/h
Fuel Level: 80%
Air Conditioning: OFF
GPS Location: Downtown

--- After Actions ---
Vehicle accelerated.
AC turned ON
GPS updated to: Uptown

--- Updated Status ---
Speed: 70 km/h
Fuel Level: 73%
Air Conditioning: ON
GPS Location: Uptown

D:\OneDrive\Desktop\OOPS\assignment\Multi1>
```

5B: MULTILEVEL INHERITANCE 2

CODE:

```
class Item {
   String title;
   String author;

   Item(String title, String author) {
      this.title = title;
      this.author = author;
   }

   void displayBasicInfo() {
      System.out.println("Title: " + title);
      System.out.println("Author: " + author);
   }
}

class Book extends Item {
   int pages;
   String genre;
```

```
    Book(String title, String author, int pages, String genre) {
        super(title, author);
        this.pages = pages;
        this.genre = genre;
    }

    void displayBookInfo() {
        displayBasicInfo();
        System.out.println("Pages: " + pages);
        System.out.println("Genre: " + genre);
    }
}

class DigitalBook extends Book {
    int fileSize;
    String quality;

    DigitalBook(String title, String author, int pages, String genre, int fileSize, String quality) {
        super(title, author, pages, genre);
        this.fileSize = fileSize;
        this.quality = quality;
    }

    void displayDigitalInfo() {
        displayBookInfo();
        System.out.println("File Size: " + fileSize + " MB");
        System.out.println("Quality: " + quality);
        System.out.println("Estimated Download Time: " + calculateDownloadTime() + " seconds");
    }

    int calculateDownloadTime() {
        int speed = 5;
        return fileSize / speed;
    }
}

public class Multi2 {
    public static void main(String[] args) {
        DigitalBook dBook = new DigitalBook("Heaven And Hell", "Deepak", 666, "Fantasy", 50, "High");
        dBook.displayDigitalInfo();
    }
}
```

OUTPUT:

```
D:\OneDrive\Desktop\OOPS\assignment\Multi2>java Multi2
Title: Heaven And Hell
Author: Deepak
Pages: 666
Genre: Fantasy
File Size: 50 MB
Quality: High
Estimated Download Time: 10 seconds
```

6A: HIERARCHICAL INHERITANCE 1

CODE:

```
class Appliance {
  String location;

  void setLocation(String location) {
    this.location = location;
  }
}

class Fan extends Appliance {
  void switchOn() {
    System.out.println("Fan in " + location + " is ON.");
  }
}

class Light extends Appliance {
  void dim(int level) {
    System.out.println("Light in " + location + " dimmed to " + level + "% brightness.");
  }
}

public class Hira1 {
  public static void main(String[] args) {
    Fan fan = new Fan();
    fan.setLocation("Living Room");
    fan.switchOn();

    Light light = new Light();
    light.setLocation("Bedroom");
```

```
        light.dim(40);
    }
}
```

OUTPUT:

```
D:\OneDrive\Desktop\OOPS\assignment\Inheritance\Hierarchical>java Hira1
Fan in Living Room is ON.
Light in Bedroom dimmed to 40% brightness.

D:\OneDrive\Desktop\OOPS\assignment\Inheritance\Hierarchical>
```

6B: HIERARCHICAL INHERITANCE 2

CODE:

```
class Account {
    String accountHolder;

    void setHolder(String name) {
        accountHolder = name;
    }
}

class Paypal extends Account {
    String email;

    void pay(double amount) {
        System.out.println(accountHolder + " paid Rs." + amount + " using PayPal (" + email + ")");
    }
}

class UPI extends Account {
    String upiId;

    void pay(double amount) {
        System.out.println(accountHolder + " paid Rs." + amount + " using UPI (" + upiId + ")");
    }
}

public class Hira2 {
    public static void main(String[] args) {
        Paypal p = new Paypal();
        p.setHolder("Deep");
        p.email = "deep@paypal.com";
```

```
        p.pay(450.75);

        UPI u = new UPI();
        u.setHolder("Deeps");
        u.upiId = "deeps@upi";
        u.pay(230.50);
    }
}
```

OUTPUT:

```
D:\OneDrive\Desktop\OOPS\assignment\Inheritance\Hierarchical>javac Hira2.java

D:\OneDrive\Desktop\OOPS\assignment\Inheritance\Hierarchical>java Hira2
Deep paid Rs.450.75 using PayPal (deep@paypal.com)
Deeps paid Rs.230.5 using UPI (deeps@upi)

D:\OneDrive\Desktop\OOPS\assignment\Inheritance\Hierarchical>
```

7A: HYBRID INHERITANCE 1

CODE:

```
class University {
  void universityDetails() {
    System.out.println("University: Amrita Vishwa Vidyapeetham, Chennai");
  }
}

class Department extends University {
  void departmentDetails() {
    System.out.println("Department: Computer Science");
  }
}

class StudentInfo {
  void studentDetails() {
    System.out.println("Student Name: Deepak");
    System.out.println("Roll No: CH.SC.U4CSE24015");
  }
}

class StudentRecord extends Department {
  StudentInfo info = new StudentInfo();
```

```
    void displayRecord() {
      universityDetails();
      departmentDetails();
      info.studentDetails();
      System.out.println("Result: Passed with Distinction");
    }
  }

  public class Hybrid {
public static void main(String[] args) {
      StudentRecord student = new StudentRecord();
      student.displayRecord();
    }
}
```

OUTPUT:

```
D:\OneDrive\Desktop\OOPS\assignment>javac Hybrid.java

D:\OneDrive\Desktop\OOPS\assignment>java Hybrid
University: Amrita Vishwa Vidyapeetham, Chennai
Department: Computer Science
Student Name: Deepak
Roll No: CH.SC.U4CSE24015
Result: Passed with Distinction

D:\OneDrive\Desktop\OOPS\assignment>
```

7B: HYBRID INHERITANCE 2

CODE:

```
class Bank {
  void bankDetails() {
    System.out.println("Bank: City Union Bank");
  }
}

class Branch extends Bank {
  void branchDetails() {
    System.out.println("Branch: Downtown City Branch");
  }
```

```
    }

    class Customer {
      void customerDetails() {
        System.out.println("Customer Name: Poojitha");
        System.out.println("Account Number: 1145780021");
      }
    }

    class Transaction extends Branch {
      Customer customer = new Customer();

      void transactionDetails() {
        bankDetails();
        branchDetails();
        customer.customerDetails();
        System.out.println("Transaction: Deposited Rs.15,000");
      }
    }

    public class Hybrid2 {
      public static void main(String[] args) {
  Transaction txn = new Transaction();
        txn.transactionDetails();
      }
}
```

OUTPUT:

```
D:\OneDrive\Desktop\OOPS\assignment\hybrid2>java Hybrid2
Bank: City Union Bank
Branch: Downtown City Branch
Customer Name: Poojitha
Account Number: 1145780021
Transaction: Deposited Rs.15,000

D:\OneDrive\Desktop\OOPS\assignment\hybrid2>
```

8A: CONSTRUCTOR OVERRIDING PROGRAM

CODE:

```
class Order {
  String customerName;
  String item;

  Order(String name, String item) {
    this.customerName = name;
    this.item = item;
  }

  void showOrder() {
    System.out.println("Customer: " + customerName);
    System.out.println("Item: " + item);
  }
}

class OnlineOrder extends Order {
  String deliveryAddress;

  OnlineOrder(String name, String item, String address) {
    super(name, item);
    this.deliveryAddress = address;
  }

  @Override
  void showOrder() {
    super.showOrder();
    System.out.println("Delivery Address: " + deliveryAddress);
  }
}

public class OR4 {
  public static void main(String[] args) {
    OnlineOrder order = new OnlineOrder("Deepak", "Wireless Keyboard", "Chennai, 600001");
    order.showOrder();
  }
}
```

OUTPUT:

```
D:\OneDrive\Desktop\OOPS\assignment\Polymorphism\OverRiding>java OR4
Customer: Deepak
Item: Wireless Keyboard
Delivery Address: Chennai, 600001

D:\OneDrive\Desktop\OOPS\assignment\Polymorphism\OverRiding>
```

9A: Constructor Overloading Program

CODE:

```
class Polynomial {
   private int[] coefficients;

   public Polynomial(int[] coefficients) {
      this.coefficients = coefficients;
   }

   public void displayPolynomial() {
      for (int i = coefficients.length - 1; i >= 0; i--) {
         if (coefficients[i] != 0) {
            if (i != coefficients.length - 1 && coefficients[i] > 0) {
               System.out.print("+");
            }
            System.out.print(coefficients[i]);
            if (i > 0) {
               System.out.print("x^" + i);
            }
            System.out.print(" ");
         }
      }
      System.out.println();
   }

   public Polynomial add(Polynomial other) {
      int maxLength = Math.max(this.coefficients.length, other.coefficients.length);
      int[] result = new int[maxLength];

      for (int i = 0; i < maxLength; i++) {
         int coef1 = (i < this.coefficients.length) ? this.coefficients[i] : 0;
         int coef2 = (i < other.coefficients.length) ? other.coefficients[i] : 0;
         result[i] = coef1 + coef2;
      }
```

```
    return new Polynomial(result);
        }

          public Polynomial multiply(Polynomial other) {
            int newDegree = this.coefficients.length + other.coefficients.length - 2;
            int[] result = new int[newDegree + 1];

            for (int i = 0; i < this.coefficients.length; i++) {
              for (int j = 0; j < other.coefficients.length; j++) {
                result[i + j] += this.coefficients[i] * other.coefficients[j];
              }
            }

            return new Polynomial(result);
          }
        }

        public class OL3 {
          public static void main(String[] args) {
            int[] poly1 = {1, 2, 3};
            Polynomial p1 = new Polynomial(poly1);
            System.out.print("Polynomial 1: ");
            p1.displayPolynomial();

            int[] poly2 = {5, 0, 3, 4};
            Polynomial p2 = new Polynomial(poly2);
            System.out.print("Polynomial 2: ");
            p2.displayPolynomial();

            Polynomial sum = p1.add(p2);
            System.out.print("Sum of Polynomials: ");
            sum.displayPolynomial();

            Polynomial product = p1.multiply(p2);
            System.out.print("Product of Polynomials: ");
    product.displayPolynomial();
        }
}
```

OUTPUT:

```
C:\Users\ASE Computer Lab\Desktop\java>java Over1
Polynomial 1: 3x^2 +2x^1 +1
Polynomial 2: 4x^3 +3x^2 +5
Sum of Polynomials: 4x^3 +6x^2 +2x^1 +6
Product of Polynomials: 12x^5 +17x^4 +10x^3 +18x^2 +10x^1 +5

C:\Users\ASE Computer Lab\Desktop\java>
```

10A: Method overloading 1

CODE:

```
class Shape {
   public double calculateArea(double radius) {
      return Math.PI * radius * radius;
   }

   public double calculateArea(double length, double breadth) {
      return length * breadth;
   }

   public double calculateArea(double base, double height, boolean isTriangle) {
      return 0.5 * base * height;
   }
}

public class OL1 {
   public static void main(String[] args) {
      Shape shape = new Shape();

      System.out.println("Area of Circle: " + shape.calculateArea(7.0));
      System.out.println("Area of Rectangle: " + shape.calculateArea(5.0, 8.0));
      System.out.println("Area of Triangle: " + shape.calculateArea(6.0, 4.0, true));
   }
}
```

OUTPUT:

```
D:\OneDrive\Desktop\OOPS\assignment\Polymorphism\OverLoading>java OL1
Area of Circle: 153.93804002589985
Area of Rectangle: 40.0
Area of Triangle: 12.0
```

10B: method overloading 2

CODE:

```
class Employee {
      public double calculateSalary(int hoursWorked, double hourlyRate) {
      return hoursWorked * hourlyRate;
   }

   public double calculateSalary(double monthlySalary) {
      return monthlySalary;
   }

   public double calculateSalary(double baseSalary, double bonus) {
      return baseSalary + bonus;
   }
}

public class OL2 {
   public static void main(String[] args) {
      Employee employee = new Employee();

      System.out.println("Hourly Employee Salary: " + employee.calculateSalary(40, 15.5));
      System.out.println("Salaried Employee Salary: " + employee.calculateSalary(3000.0));
      System.out.println("Employee with Bonus Salary: " + employee.calculateSalary(2500.0, 500.0));
   }
}
```

OUTPUT:

```
C:\Users\ASE Computer Lab\Desktop\java>java Over1
Hourly Employee Salary: 620.0
Salaried Employee Salary: 3000.0
Employee with Bonus Salary: 3000.0

C:\Users\ASE Computer Lab\Desktop\java>
```

11A: method overriding 1

CODE:

```
class BankAccount {
   protected String accountNumber;
   protected double balance;

   public BankAccount(String accountNumber, double balance) {
      this.accountNumber = accountNumber;
      this.balance = balance;
   }

   public void deposit(double amount) {
      balance += amount;
      System.out.println("Deposited: $" + amount);
   }

   public void withdraw(double amount) {
      if(amount <= balance) {
         balance -= amount;
         System.out.println("Withdrawn: $" + amount);
      } else {
```

```
            System.out.println("Insufficient funds!");
        }
    }

    public void displayBalance() {
        System.out.println("Current Balance: $" + balance);
    }
}

class SavingsAccount extends BankAccount {
    private double interestRate;

    public SavingsAccount(String accountNumber, double balance, double interestRate) {
        super(accountNumber, balance);
        this.interestRate = interestRate;
    }

    public void addInterest() {
        double interest = balance * interestRate;
        balance += interest;
        System.out.println("Interest added: $" + interest);
    }

    public void displayBalance() {
        System.out.println("Savings Account Balance: $" + balance);
        System.out.println("Interest Rate: " + (interestRate * 100) + "%");
    }
}

class CheckingAccount extends BankAccount {
    private double overdraftLimit;

    public CheckingAccount(String accountNumber, double balance, double overdraftLimit) {
        super(accountNumber, balance);
        this.overdraftLimit = overdraftLimit;
    }

    public void withdraw(double amount) {
        if(amount <= (balance + overdraftLimit)) {
            balance -= amount;
            System.out.println("Withdrawn: $" + amount);
        } else {
            System.out.println("Exceeds overdraft limit!");
        }
    }

    public void displayBalance() {
        System.out.println("Checking Account Balance: $" + balance);
        System.out.println("Overdraft Limit: $" + overdraftLimit);
    }
}

public class OR2 {
    public static void main(String[] args) {
        BankAccount savings = new SavingsAccount("SA001", 1000.0, 0.05);
        BankAccount checking = new CheckingAccount("CA001", 2000.0, 500.0);

        System.out.println("Initial Account Status:");
        System.out.println("----------------------");
        savings.displayBalance();
        System.out.println();
```

```
        checking.displayBalance();

        System.out.println("\nPerforming Transactions:");
        System.out.println("----------------------");
        savings.deposit(500.0);
        checking.withdraw(2300.0);

        ((SavingsAccount)savings).addInterest();

        System.out.println("\nFinal Account Status:");
        System.out.println("--------------------");
        savings.displayBalance();
        System.out.println();
        checking.displayBalance();
    }
}
```

OUTPUT:

```
Initial Account Status:
-----------------------
Savings Account Balance: $1000.0
Interest Rate: 5.0%

Checking Account Balance: $2000.0
Overdraft Limit: $500.0

Performing Transactions:
----------------------
Deposited: $500.0
Withdrawn: $2300.0
Interest added: $75.0

Final Account Status:
--------------------
Savings Account Balance: $1575.0
Interest Rate: 5.0%

Checking Account Balance: $-300.0
Overdraft Limit: $500.0
```

11B: method overriding 2

CODE:

```
class Shape {
   public double getArea() {
      return 0.0;
   }

   public void draw() {
      System.out.println("Drawing a shape");
   }
}

class Circle extends Shape {
   private double radius;

   public Circle(double radius) {
      this.radius = radius;
   }

   @Override
   public double getArea() {
      return Math.PI * radius * radius;
   }

   @Override
   public void draw() {
      System.out.println("Drawing a circle with radius: " + radius);
   }
}

class Rectangle extends Shape {
   private double length;
   private double width;

   public Rectangle(double length, double width) {
      this.length = length;
      this.width = width;
   }

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

   @Override
   public void draw() {
      System.out.println("Drawing a rectangle with length: " + length + " and width: " + width);
   }
}

public class OR1 {
   public static void main(String[] args) {
      Shape circle = new Circle(5);
      Shape rectangle = new Rectangle(4, 6);

      System.out.println("Shape Areas:");
      System.out.println("Circle area: " + circle.getArea());
      System.out.println("Rectangle area: " + rectangle.getArea());

      System.out.println("\nDrawing Shapes:");
```

```
        circle.draw();
        rectangle.draw();

        Shape[] shapes = {new Circle(3), new Rectangle(2, 5)};

        System.out.println("\nProcessing shapes array:");
        for(Shape shape : shapes) {
            shape.draw();
            System.out.println("Area: " + shape.getArea());
        }
    }
}
```

OUTPUT:

```
D:\OneDrive\Desktop\OOPS\assignment\Polymorphism\OverRiding>java OR1
Shape Areas:
Circle area: 78.53981633974483
Rectangle area: 24.0

Drawing Shapes:
Drawing a circle with radius: 5.0
Drawing a rectangle with length: 4.0 and width: 6.0

Processing shapes array:
Drawing a circle with radius: 3.0
Area: 28.274333882308138
Drawing a rectangle with length: 2.0 and width: 5.0
Area: 10.0
```

12A: interface program 1

CODE:

```
interface Shape {
    double area();
    double perimeter();
}

class Circle implements Shape {
    private double radius;

    public Circle(double radius) {
        this.radius = radius;
    }

    public double area() {
        return Math.PI * radius * radius;
    }

    public double perimeter() {
        return 2 * Math.PI * radius;
    }
}

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

```
    public Rectangle(double length, double width) {
        this.length = length;
        this.width = width;
    }

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

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

class Triangle implements Shape {
    private double a, b, c;

    public Triangle(double a, double b, double c) {
        this.a = a;
        this.b = b;
        this.c = c;
    }

    public double area() {
        double s = perimeter() / 2;
        return Math.sqrt(s * (s - a) * (s - b) * (s - c));
    }

    public double perimeter() {
        return a + b + c;
    }
}

public class Inter1 {
    public static void main(String[] args) {
        Shape circle = new Circle(5);
        Shape rectangle = new Rectangle(4, 6);
        Shape triangle = new Triangle(3, 4, 5);

        System.out.println("\nCircle Area: " + circle.area() + ", Perimeter: " + circle.perimeter());
        System.out.println("Rectangle Area: " + rectangle.area() + ", Perimeter: " + rectangle.perimeter());
        System.out.println("Triangle Area: " + triangle.area() + ", Perimeter: " + triangle.perimeter());
    }
}
```

OUTPUT:

```
D:\OneDrive\Desktop\OOPS\assignment\Abstraction\Interface>java Inter1

Circle Area: 78.53981633974483, Perimeter: 31.41592653589793
Rectangle Area: 24.0, Perimeter: 20.0
Triangle Area: 6.0, Perimeter: 12.0
```

12B: interface program 2

CODE:

```
interface PaymentMethod {
   boolean processPayment(double amount);
}

class CreditCard implements PaymentMethod {
   private String cardNumber;

   public CreditCard(String cardNumber) {
      this.cardNumber = cardNumber;
   }

   public boolean processPayment(double amount) {
      System.out.println("Processing credit card payment of Rs : " + amount);
      return true;
   }
}

class GooglePay implements PaymentMethod {
   private String email;

   public GooglePay(String email) {
      this.email = email;
   }

   public boolean processPayment(double amount) {
      System.out.println("Processing GooglePay payment of Rs : " + amount);
      return true;
   }
}

class BankTransfer implements PaymentMethod {
   private String bankAccount;

   public BankTransfer(String bankAccount) {
      this.bankAccount = bankAccount;
}

   public boolean processPayment(double amount) {
      System.out.println("Processing bank transfer of Rs : " + amount);
      return true;
   }
}

public class Inter2 {
   public static void main(String[] args) {
      PaymentMethod paymentMethod1 = new CreditCard("1234-5678-9876-5432");
      PaymentMethod paymentMethod2 = new GooglePay("upi@gmail.com");
      PaymentMethod paymentMethod3 = new BankTransfer("123456789");
            System.out.println();
      paymentMethod1.processPayment(1025.245);
      paymentMethod2.processPayment(986.53);
      paymentMethod3.processPayment(84543.25);
   }
}
```

OUTPUT:

```
D:\OneDrive\Desktop\OOPS\assignment\Abstraction\Interface>java Inter2

Processing credit card payment of Rs : 1025.245
Processing GooglePay payment of Rs : 986.53
Processing bank transfer of Rs : 84543.25

D:\OneDrive\Desktop\OOPS\assignment\Abstraction\Interface>
```

12C: interface program 3

CODE:

```
import java.util.*;

interface LibraryItem {
   String getDetails();
   boolean borrow();
   void returnItem();
}

class Book implements LibraryItem {
   private String title;
   private String author;
   private boolean isBorrowed;

   public Book(String title, String author) {
      this.title = title;
      this.author = author;
      this.isBorrowed = false;
   }

   public String getDetails() {
      return "Book: " + title + " by " + author;
   }

   public boolean borrow() {
      if (!isBorrowed) {
         isBorrowed = true;
         System.out.println("You borrowed the book: " + title);
         return true;
      } else {
         System.out.println("The book is already borrowed.");
         return false;
      }
   }

   public void returnItem() {
      isBorrowed = false;
      System.out.println("You returned the book: " + title);
   }
}

class Library {
   private List<LibraryItem> items = new ArrayList<>();

   public void addItem(LibraryItem item) {
      items.add(item);
   }
```

```
    public void showItems() {
        for (LibraryItem item : items) {
            System.out.println(item.getDetails());
        }
    }

    public LibraryItem getItem(String title) {
        for (LibraryItem item : items) {
            if (item.getDetails().contains(title)) {
                return item;
            }
        }
        return null;
    }
}

public class Inter3 {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        Library library = new Library();

        library.addItem(new Book("Heaven and Hell", "Deepak"));
        library.addItem(new Book("2025", "Vaishnav"));

        while (true) {
            System.out.println("\nLibrary System: ");
            System.out.println("1. View Library Items");
            System.out.println("2. Borrow Book");
            System.out.println("3. Return Book");
            System.out.println("4. Exit");
            System.out.print("Enter your choice: ");
            int choice = scanner.nextInt();
            scanner.nextLine();

            switch (choice) {
                case 1:
                    library.showItems();
                    break;
                case 2:
                    System.out.print("Enter the book title you want to borrow: ");
                    String borrowTitle = scanner.nextLine();
                    LibraryItem itemToBorrow = library.getItem(borrowTitle);
                    if (itemToBorrow != null) {
                        itemToBorrow.borrow();
                    } else {
                        System.out.println("Book not found.");
                    }
                    break;
                case 3:
                    System.out.print("Enter the book title you want to return: ");
                    String returnTitle = scanner.nextLine();
                    LibraryItem itemToReturn = library.getItem(returnTitle);
                    if (itemToReturn != null) {
                        itemToReturn.returnItem();
                    } else {
                        System.out.println("Book not found." );
                    }
                    break;
                case 4:
                    System.out.println("Exiting system...");
```

```
            return;
          default:
            System.out.println("Invalid choice, please try again.");
        }
      }
    }
}
```

OUTPUT:

```
Library System:
1. View Library Items
2. Borrow Book
3. Return Book
4. Exit
Enter your choice: 1
Book: Heaven and Hell by Deepak
Book: 2025 by Vaishnav

Library System:
1. View Library Items
2. Borrow Book
3. Return Book
4. Exit
Enter your choice: 2
Enter the book title you want to borrow: Heaven and Hell
You borrowed the book: Heaven and Hell

Library System:
1. View Library Items
2. Borrow Book
3. Return Book
4. Exit
Enter your choice: 3
Enter the book title you want to return: Heaven and Hell
You returned the book: Heaven and Hell

Library System:
1. View Library Items
2. Borrow Book
3. Return Book
4. Exit
Enter your choice: 4
Exiting system...

D:\OneDrive\Desktop\OOPS\assignment\Abstraction\Interface>
```

12D: interface program 4

CODE:

```
import java.util.*;

interface Task {
   void start();
   void pause();
   void complete();
}

class AppTask implements Task {
   public void start() {
      System.out.println("Started scanning apps.");
   }

   public void pause() {
      System.out.println("Paused scanning apps.");
   }

   public void complete() {
      System.out.println("Completed scanning apps.");
   }
}

class VirusTask implements Task {
   public void start() {
      System.out.println("Started scanning for virus.");
   }

   public void pause() {
      System.out.println("Paused scanning for virus.");
   }

   public void complete() {
      System.out.println("Completed scanning for virus.");
   }
}
class TrojanTask implements Task {
   public void start() {
      System.out.println("Started scanning for trojans");
   }

   public void pause() {
      System.out.println("Paused scanning for trojans.");
   }

   public void complete() {
      System.out.println("Completed scanning for trojans.");
   }
}

public class Inter4 {
   public static void main(String[] args) {
      Scanner scanner = new Scanner(System.in);
      List<Task> tasks = new ArrayList<>();
      tasks.add(new AppTask());
      tasks.add(new VirusTask());
      tasks.add(new TrojanTask());

      while (true) {
```

```
        System.out.println("\nTask Manager:");
        System.out.println("1. Start App Task");
        System.out.println("2. Pause Virus Task");
        System.out.println("3. Complete Trojan Task");
        System.out.println("4. Exit");
        System.out.print("Enter your choice: ");
        int choice = scanner.nextInt();

        switch (choice) {
          case 1:
            tasks.get(0).start();
           break;
          case 2:
            tasks.get(1).pause();
            break;
          case 3:
            tasks.get(2).complete();
            break;
          case 4:
            System.out.println("Exiting Task Manager.");
            return;
          default:
            System.out.println("Invalid choice. Try again.");
        }
      }
    }
}
```

OUTPUT:

```
Task Manager:
1. Start App Task
2. Pause Virus Task
3. Complete Trojan Task
4. Exit
Enter your choice: 1
Started scanning apps.

Task Manager:
1. Start App Task
2. Pause Virus Task
3. Complete Trojan Task
4. Exit
Enter your choice: 2
Paused scanning for virus.

Task Manager:
1. Start App Task
2. Pause Virus Task
3. Complete Trojan Task
4. Exit
Enter your choice: 3
Completed scanning for trojans.

Task Manager:
1. Start App Task
2. Pause Virus Task
3. Complete Trojan Task
4. Exit
Enter your choice: 4
Exiting Task Manager.

D:\OneDrive\Desktop\OOPS\assignment\Abstraction\Interface>
```

13A: abstract program 1

CODE:

```
abstract class Abs1 {
   abstract void Info();
}

class Management extends Abs1 {
   void Info() {
      String name = "Deepak";
      String rollno = "CH.SC.U4CSE24015";
      int age = 18;
      String gender = "Male";

      System.out.println("The name of the student is: " + name);
      System.out.println("The roll number of the student is: " + rollno);
      System.out.println("The age of the student is: " + age);
      System.out.println("The gender of the student is: " + gender);
   }
}

public class Abase1 {
   public static void main(String args[]) {
      Abs1 s = new Management();
      s.Info();
   }
}
```

OUTPUT:

```
D:\OneDrive\Desktop\OOPS\assignment\Abstraction\Abstract>java Abase1
The name of the student is: Deepak
The roll number of the student is: CH.SC.U4CSE24015
The age of the student is: 18
The gender of the student is: Male
```

13B: abstract program 2

CODE:

```
abstract class Subject {
   Subject() {
     System.out.println("Subjects : ");
   }

     abstract void syllabus();

     void Learn(){
        System.out.println("Learning!!!");
   }
}

class College extends Subject {
  void syllabus(){
   System.out.println("Java , HTML , Python");
  }
}

class Abase2 {
   public static void main(String[] args) {
      Subject s = new College();
```

```
        s.syllabus();
        s.Learn();
    }
}
```

OUTPUT:

```
D:\OneDrive\Desktop\OOPS\assignment\Abstraction\Abstract>java Abase2
Subjects :
Java , HTML , Python
Learning!!!

D:\OneDrive\Desktop\OOPS\assignment\Abstraction\Abstract>
```

13C: abstract program 3

CODE:

```
import java.util.Scanner;

abstract class B {
    static abstract class C {
        abstract void myAbstractMethod();
    }
}

class D extends B {
    class E extends C {
        Scanner scan = new Scanner(System.in);

        void myAbstractMethod() {
            System.out.print("Enter your name: ");
            String name = scan.nextLine();

            System.out.print("Enter your roll number: ");
            String rollno = scan.nextLine();

            System.out.print("Enter your age: ");
            int age = scan.nextInt();

            System.out.println("\nYour name is: " + name);
            System.out.println("Your roll number is: " + rollno);
            System.out.println("Your age is: " + age);
        }
    }
}

public class Abase3 {
    public static void main(String[] args) {
        D outer = new D();
        D.E inner = outer.new E();
        inner.myAbstractMethod();
    }
}
```

OUTPUT:

```
D:\OneDrive\Desktop\OOPS\assignment\Abstraction\Abstract>java Abase3
Enter your name: Deepak
Enter your roll number: CH.SC.U4CSE24015
Enter your age: 18

Your name is: Deepak
Your roll number is: CH.SC.U4CSE24015
Your age is: 18
```

13D: abstract program 4

CODE:

```
import java.util.Scanner;

abstract class Movie{
abstract void hero();
abstract void antiHero();
abstract void villain();
}

abstract class character extends Movie{
Scanner scan = new Scanner(System.in);
public void hero(){
System.out.print("Enter a quote for the hero : ");
String hero = scan.nextLine();
System.out.println("Quote of the hero is : "+hero);
}
}

class character2 extends character{
void antiHero(){

System.out.print("Enter a quote for the anti-hero : ");
String antiHero = scan.nextLine();
System.out.println("Quote of the anti-hero is : "+antiHero);
}

void villain(){

System.out.print("Enter a quote for the villain : ");
String villain = scan.nextLine();
System.out.println("Quote of the villain is : "+villain);
}
}

class Abase4 {
public static void main(String[] args) {
character2 c = new character2();
c.hero();
System.out.println("-------------------------------------");
c.antiHero();
System.out.println("-------------------------------------");
c.villain();
System.out.println("-------------------------------------");
}
}
```

OUTPUT:

```
D:\OneDrive\Desktop\OOPS\assignment\Abstraction\Abstract>java Abase4
Enter a quote for the hero : "By the people , for the people , of the people"
Quote of the hero is : "By the people , for the people , of the people"
-----------------------------------
Enter a quote for the anti-hero : "Do what satisfies you no matter its good or bad"
Quote of the anti-hero is : "Do what satisfies you no matter its good or bad"
-----------------------------------
Enter a quote for the villain : "The one who kills others is not a villain but a God who cleans up the trash"
Quote of the villain is : "The one who kills others is not a villain but a God who cleans up the trash"
-----------------------------------
```

14A: encapsulation 1

CODE:

```
class BankAccount {
   private String accountHolder;
   private double balance;

   public BankAccount(String accountHolder, double initialBalance) {
      this.accountHolder = accountHolder;
      this.balance = initialBalance;
   }

   public String getAccountHolder() {
      return accountHolder;
   }

   public double getBalance() {
      return balance;
   }

   public void deposit(double amount) {
      if (amount > 0) {
         balance += amount;
         System.out.println("Deposited: " + amount);
      } else {
         System.out.println("Invalid deposit amount.");
      }
   }

   public void withdraw(double amount) {
      if (amount > 0 && balance >= amount) {
         balance -= amount;
         System.out.println("Withdrawn: " + amount);
      } else {
         System.out.println("Invalid withdrawal amount or insufficient funds.");
      }
   }


   public void displayAccountInfo() {
      System.out.println("Account Holder: " + accountHolder);
      System.out.println("Current Balance: " + balance);
   }
}

public class En1 {
   public static void main(String[] args) {
      BankAccount account = new BankAccount("Ben Dover", 10000);
      account.displayAccountInfo();
      account.deposit(345);
```

```
    account.withdraw(6736);
    account.displayAccountInfo();
  }
}
```

OUTPUT:

```
D:\OneDrive\Desktop\OOPS\assignment\Encapsulation>javac En1.java

D:\OneDrive\Desktop\OOPS\assignment\Encapsulation>java En1
Account Holder: Ben Dover
Current Balance: 10000.0
Deposited: 345.0
Withdrawn: 6736.0
Account Holder: Ben Dover
Current Balance: 3609.0
```

14B: encapsulation 2

CODE:

```
class Employee {
  private String name;
  private String position;
  private double salary;

  public Employee(String name, String position, double salary) {
    this.name = name;
    this.position = position;
    this.salary = salary;
  }

  public String getName() {
    return name;
  }

  public String getPosition() {
    return position;
  }

  public double getSalary() {
    return salary;
  }

  public void giveRaise(double raiseAmount) {
    if (raiseAmount > 0) {
      salary += raiseAmount;
      System.out.println("Salary raised by: " + raiseAmount);
    } else {
      System.out.println("Invalid raise amount.");
    }
  }

  public void displayEmployeeInfo() {
    System.out.println("Employee Name: " + name);
    System.out.println("Position: " + position);
    System.out.println("Salary: " + salary);
  }
}
```

```
public class En2 {
   public static void main(String[] args) {
      Employee emp = new Employee("Vaishnav", "Civil Engineer", 6000);
      emp.displayEmployeeInfo();
      emp.giveRaise(2500);
      emp.displayEmployeeInfo();
   }
}
```

OUTPUT:

```
D:\OneDrive\Desktop\OOPS\assignment\Encapsulation>java En2
Employee Name: Vaishnav
Position: Civil Engineer
Salary: 6000.0
Salary raised by: 2500.0
Employee Name: Vaishnav
Position: Civil Engineer
Salary: 8500.0
```

14C: encapsulation 3

CODE:

```
import java.util.Scanner;

class BankAccount {
   private String accountHolder;
   private double balance;

   public BankAccount(String accountHolder, double initialBalance) {
      this.accountHolder = accountHolder;
      this.balance = initialBalance;
   }

   public String getAccountHolder() {
      return accountHolder;
   }

   public double getBalance() {
      return balance;
   }

   public void deposit(double amount) {
      if (amount > 0) {
         balance += amount;
         System.out.println("Deposited: $" + amount);
      } else {
         System.out.println("Invalid deposit amount.");
      }
   }

   public void withdraw(double amount) {
      if (amount > 0 && balance >= amount) {
         balance -= amount;
         System.out.println("Withdrawn: $" + amount);
      } else {
```

```
            System.out.println("Insufficient funds or invalid amount.");
        }
    }

    public void displayAccountInfo() {
        System.out.println("Account Holder: " + accountHolder);
        System.out.println("Balance: $" + balance);
    }
}

class Book {
    private String title;
    private String author;
    private boolean isAvailable;

    public Book(String title, String author) {
        this.title = title;
        this.author = author;
        this.isAvailable = true;
    }

    public void checkOut() {
        if (isAvailable) {
            isAvailable = false;
            System.out.println("Book '" + title + "' has been checked out.");
        } else {
            System.out.println("Sorry, the book '" + title + "' is not available.");
        }
    }

    public void returnBook() {
        if (!isAvailable) {
            isAvailable = true;
            System.out.println("Book '" + title + "' has been returned.");
        } else {
            System.out.println("This book was not checked out.");
        }
    }

    public void displayBookInfo() {
        System.out.println("Title: " + title);
        System.out.println("Author: " + author);
        System.out.println("Available: " + (isAvailable ? "Yes" : "No"));
    }
}

class Student {
    private String name;
    private double[] grades;
    private int gradeCount;

    public Student(String name, int maxGrades) {
        this.name = name;
        grades = new double[maxGrades];
        gradeCount = 0;
    }

    public void addGrade(double grade) {
        if (gradeCount < grades.length) {
            grades[gradeCount++] = grade;
        } else {
```

```
            System.out.println("Cannot add more grades.");
        }
    }

    public double calculateAverage() {
        double sum = 0;
        for (int i = 0; i < gradeCount; i++) {
            sum += grades[i];
        }
        return (gradeCount > 0) ? sum / gradeCount : 0;
    }

    public void displayStudentInfo() {
        System.out.println("Student: " + name);
        System.out.println("Average Grade: " + calculateAverage());
    }
}

class Employee {
    private String name;
    private String position;
    private double salary;

    public Employee(String name, String position, double salary) {
        this.name = name;
        this.position = position;
        this.salary = salary;
    }

    public void giveRaise(double amount) {
        if (amount > 0) {
            salary += amount;
            System.out.println("Salary raised by: $" + amount);
        } else {
            System.out.println("Invalid raise amount.");
        }
    }

    public void displayEmployeeInfo() {
        System.out.println("Employee: " + name);
        System.out.println("Position: " + position);
        System.out.println("Salary: $" + salary);
    }
}

public class En3 {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        BankAccount account = new BankAccount("Ben Dover", 20000);
        Book book = new Book("Heaven and Hell", "Deepak");
        Student student = new Student("Vaishnav", 5);
        Employee employee = new Employee("Sanjay", "Detective", 50000);

        while (true) {
            System.out.println("\nSelect an option:");
            System.out.println("1. Bank Account");
            System.out.println("2. Library System");
            System.out.println("3. Student Grades");
            System.out.println("4. Employee Management");
            System.out.println("5. Exit");
            System.out.print("Choice: ");
```

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

switch (choice) {
   case 1:
      System.out.println("\nBank Account Operations:");
      System.out.println("1. Deposit");
      System.out.println("2. Withdraw");
      System.out.println("3. Display Info");
      System.out.print("Choice: ");
      int bankChoice = sc.nextInt();
      switch (bankChoice) {
         case 1:
            System.out.print("Enter deposit amount: ");
            double deposit = sc.nextDouble();
            account.deposit(deposit);
            break;
         case 2:
            System.out.print("Enter withdrawal amount: ");
            double withdrawal = sc.nextDouble();
            account.withdraw(withdrawal);
            break;
         case 3:
            account.displayAccountInfo();
            break;
         default:
            System.out.println("Invalid choice.");
      }
      break;

   case 2:
      System.out.println("\nLibrary System Operations:");
      System.out.println("1. Check out Book");
      System.out.println("2. Return Book");
      System.out.println("3. Display Book Info");
      System.out.print("Choice: ");
      int bookChoice = sc.nextInt();
      switch (bookChoice) {
         case 1:
            book.checkOut();
            break;
         case 2:
            book.returnBook();
            break;
         case 3:
            book.displayBookInfo();
            break;
         default:
            System.out.println("Invalid choice.");
      }
      break;

   case 3:
      System.out.println("\nStudent Grades Operations:");
      System.out.println("1. Add Grade");
      System.out.println("2. Display Student Info");
      System.out.print("Choice: ");
      int studentChoice = sc.nextInt();
      switch (studentChoice) {
         case 1:
            System.out.print("Enter grade: ");
            double grade = sc.nextDouble();
```

```
                student.addGrade(grade);
                break;
            case 2:
                student.displayStudentInfo();
                break;
            default:
                System.out.println("Invalid choice.");
            }
            break;

        case 4:
            System.out.println("\nEmployee Management Operations:");
            System.out.println("1. Give Raise");
            System.out.println("2. Display Employee Info");
            System.out.print("Choice: ");
            int empChoice = sc.nextInt();
            switch (empChoice) {
                case 1:
                    System.out.print("Enter raise amount: ");
                    double raise = sc.nextDouble();
                    employee.giveRaise(raise);
                    break;
                case 2:
                    employee.displayEmployeeInfo();
                    break;
                default:
                    System.out.println("Invalid choice.");
            }
            break;

        case 5:
            System.out.println("Exiting...");
            sc.close();
            return;

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

OUTPUT:

```
Select an option:
1. Bank Account
2. Library System
3. Student Grades
4. Employee Management
5. Exit
Choice: 1

Bank Account Operations:
1. Deposit
2. Withdraw
3. Display Info
Choice: 1
Enter deposit amount: 5000
Deposited: $5000.0

Select an option:
1. Bank Account
2. Library System
3. Student Grades
4. Employee Management
5. Exit
Choice: 2

Library System Operations:
1. Check out Book
2. Return Book
3. Display Book Info
Choice: 1
Book 'Heaven and Hell' has been checked out.

Select an option:
1. Bank Account
2. Library System
3. Student Grades
4. Employee Management
5. Exit
Choice: 4

Employee Management Operations:
1. Give Raise
2. Display Employee Info
Choice: 1
Enter raise amount: 1000
Salary raised by: $1000.0

Select an option:
1. Bank Account
2. Library System
3. Student Grades
4. Employee Management
5. Exit
Choice: 5
Exiting...

D:\OneDrive\Desktop\OOPS\assignment\Encapsulation>
```

14D: encapsulation 4

CODE:

```
abstract class Shape {
   public abstract double area();
}

class Person {
   private String name;
   private int age;

   public Person(String name, int age) {
      this.name = name;
      this.age = age;
   }

   public int getAge() {
      return age;
   }

   public void setAge(int age) {
      if (age > 0) {
         this.age = age;
      } else {
         System.out.println("Age cannot be negative");
      }
   }

   public void introduce() {
      System.out.println("Hi, I am " + name + " and I am " + age + " years old.");
   }
}

class Employee extends Person {
   private String position;

   public Employee(String name, int age, String position) {
      super(name, age);
      this.position = position;
   }

   public void introduce() {
      System.out.println("Hi, I am " + super.getAge() + " years old. I work as a " + position + ".");
   }
}

class Circle extends Shape {
   private double radius;

   public Circle(double radius) {
      this.radius = radius;
   }

   public double area() {
      return 3.14 * radius * radius;
   }
}

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

```
    public Rectangle(double length, double width) {
        this.length = length;
        this.width = width;
    }

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


class Calculator {
    public int add(int a, int b) {
        return a + b;
    }

    public int add(int a, int b, int c) {
        return a + b + c;
    }
}

public class En4 {
    public static void main(String[] args) {
        Person person = new Person("Raj", 29);
        Employee employee = new Employee("Sanjay", 21, "Civil Engineer");

        person.introduce();
        employee.introduce();
        System.out.println("Employee's age is: " + employee.getAge());
        employee.setAge(26);
        System.out.println("Updated Employee's age: " + employee.getAge());

        Shape circle = new Circle(5);
        Shape rectangle = new Rectangle(10, 4);
        System.out.println("Area of circle: " + circle.area());
        System.out.println("Area of rectangle: " + rectangle.area());

        Calculator calculator = new Calculator();
        System.out.println("Sum of 10 and 20: " + calculator.add(10, 20));
        System.out.println("Sum of 10, 20, and 30: " + calculator.add(10, 20, 30));
    }
}
```

OUTPUT:

```
C:\Users\ASE Computer Lab\Desktop\java>java Encap
Hi, I am Raj and I am 29 years old.
Hi, I am 21 years old. I work as a Civil Engineer.
Employee's age is: 21
Updated Employee's age: 26
Area of circle: 78.5
Area of rectangle: 40.0
Sum of 10 and 20: 30
Sum of 10, 20, and 30: 60

C:\Users\ASE Computer Lab\Desktop\java>
```

15A: user-defined package 1

CODE:

```
package pkg;

public class Rectangle {
   private double length;
   private double breadth;

   public Rectangle(double l, double b) {
      length = l;
      breadth = b;
   }

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

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

main program :

```
import pkg.Rectangle;

public class Userpk1 {
   public static void main(String[] args) {
      Rectangle r = new Rectangle(10.5, 5.2);
      System.out.println("Area of Rectangle: " + r.getArea());
      System.out.println("Perimeter of Rectangle: " + r.getPerimeter());
   }
}
```

OUTPUT:

```
D:\OneDrive\Desktop\OOPS\assignment\user package1>java Userpk1
Area of Rectangle: 54.6
Perimeter of Rectangle: 31.4
```

15B: user-defined package 2

CODE:

```
package pkg;

public class BMICalculator {
   private double height;
   private double weight;
   public BMICalculator(double height, double weight) {
      this.height = height;
      this.weight = weight;
   }

   public double calculateBMI() {
      return (weight / (height * height));
   }

   public String getCategory() {
      double bmi = calculateBMI();
```

```
        if (bmi < 18.5) {
            return "Underweight";
        } else if (bmi < 24.9) {
            return "Normal weight";
        } else if (bmi < 29.9) {
            return "Overweight";
        } else {
            return "Obese";
        }
    }
}
```

main program :

```
import pkg.BMICalculator;
import java.util.Scanner;

public class Userpk2 {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        System.out.print("Enter your height in meters: ");
        double height = scanner.nextDouble();

        System.out.print("Enter your weight in kilograms: ");
        double weight = scanner.nextDouble();

        BMICalculator calculator = new BMICalculator(height, weight);
        double bmi = calculator.calculateBMI();
        String category = calculator.getCategory();

        System.out.printf("Your BMI is: %.2f\n", bmi);
        System.out.println("You are classified as: " + category);
    }
}
```

OUTPUT:

```
D:\OneDrive\Desktop\OOPS\assignment\user package1>java Userpk2
Enter your height in meters: 2
Enter your weight in kilograms: 2
Your BMI is: 0.50
You are classified as: Underweight
```

15C: in-built package 3

CODE:

```
import java.io.FileWriter;
import java.io.IOException;
import java.net.InetAddress;
import java.net.UnknownHostException;
import java.time.LocalDateTime;
import java.time.format.DateTimeFormatter;
import java.util.Scanner;

public class Package1 {
   public static void main(String[] args) {
      Scanner scanner = new Scanner(System.in);
      System.out.println("Enter website (e.g., www.example.com):");
      String website = scanner.nextLine();

      try {
         InetAddress inet = InetAddress.getByName(website);
         String ip = inet.getHostAddress();

         LocalDateTime timestamp = LocalDateTime.now();
         DateTimeFormatter formatter = DateTimeFormatter.ofPattern("yyyy-MM-dd HH:mm:ss");

         String log = "Website: " + website + "\nIP Address: " + ip +
                  "\nTimestamp: " + timestamp.format(formatter) + "\n\n";

         FileWriter writer = new FileWriter("log.txt", true);
         writer.write(log);
         writer.close();

         System.out.println("Website info logged successfully!");
      } catch (UnknownHostException e) {
         System.out.println("Website not found!");
      } catch (IOException e) {
         System.out.println("Error writing to file.");
      }
   }
}
```

OUTPUT:

```
D:\OneDrive\Desktop\OOPS\assignment>javac Package1.java

D:\OneDrive\Desktop\OOPS\assignment>java Package1
Enter website (e.g., www.example.com):
www.google.com
Website info logged successfully!

D:\OneDrive\Desktop\OOPS\assignment>
```

15D: in-built package 4

CODE:

```
import java.io.FileWriter;
import java.io.IOException;
import java.net.InetAddress;
import java.time.LocalDateTime;
import java.time.format.DateTimeFormatter;

class Attendance {
   private String studentName;
   private String ipAddress;
   private String loginTime;

   public void setStudentName(String name) {
      this.studentName = name;
   }

   public String getStudentName() {
      return studentName;
   }

   public void setIpAddress(String ip) {
      this.ipAddress = ip;
   }

   public String getIpAddress() {
      return ipAddress;
   }

   public void setLoginTime(String time) {
      this.loginTime = time;
   }

   public String getLoginTime() {
      return loginTime;
   }

   public String getFormattedRecord() {
      return "Name: " + studentName + ", IP: " + ipAddress + ", Login Time: " + loginTime + "\n";
   }
}

public class Package2 {
   public static void main(String[] args) {
      Attendance a = new Attendance();
      a.setStudentName("Rahul Sharma");

      try {
         InetAddress inet = InetAddress.getLocalHost();
         a.setIpAddress(inet.getHostAddress());

         LocalDateTime now = LocalDateTime.now();
         DateTimeFormatter dtf = DateTimeFormatter.ofPattern("yyyy-MM-dd HH:mm:ss");
         a.setLoginTime(dtf.format(now));

         FileWriter writer = new FileWriter("attendance_log.txt", true);
         writer.write(a.getFormattedRecord());
         writer.close();

         System.out.println("Attendance logged successfully.");
```

```
    } catch (IOException e) {
      System.out.println("Error: " + e.getMessage());
    }
  }
}
```

OUTPUT:

```
D:\OneDrive\Desktop\OOPS\assignment\Package2>javac Package2.java

D:\OneDrive\Desktop\OOPS\assignment\Package2>java Package2
Attendance logged successfully.

D:\OneDrive\Desktop\OOPS\assignment\Package2>
```

16A: exception handling 1

CODE:

```
public class Excp1 {
  public static void main(String[] args) {
    String s = "abc";
    try {
      int num = Integer.parseInt(s);
      System.out.println("Number: " + num);
    } catch (NumberFormatException e) {
      System.out.println("Invalid number format!");
    }
  }
}
```

OUTPUT:

```
D:\OneDrive\Desktop\OOPS\assignment>javac Excp1.java

D:\OneDrive\Desktop\OOPS\assignment>java Excp1
Invalid number format!

D:\OneDrive\Desktop\OOPS\assignment>
```

16B: exception handling 2

CODE:

```
import java.util.Scanner;

public class Excp2 {
  public static void main(String[] args) {
    Scanner sc = new Scanner(System.in);
    String correctUsername = "admin";
    String correctPassword = "1234";

    try {
      System.out.print("Enter username: ");
      String username = sc.nextLine();

      System.out.print("Enter password: ");
      String password = sc.nextLine();
```

```
        if (!username.equals(correctUsername) || !password.equals(correctPassword)) {
            throw new Exception("Invalid username or password.");
        }

        System.out.println("Login Successful!");

    } catch (Exception e) {
        System.out.println("Login failed: " + e.getMessage());
    }
  }
}
```

OUTPUT:

```
D:\OneDrive\Desktop\OOPS\assignment>java Excp2
Enter username: deepak
Enter password: 123
Login failed: Invalid username or password.

D:\OneDrive\Desktop\OOPS\assignment>java Excp2
Enter username: admin
Enter password: 1234
Login Successful!
```

16C: exception handling 3

CODE:

```
import java.io.*;

public class Excp3 {
  public static void main(String[] args) {
    BufferedReader reader = null;

    try {
      reader = new BufferedReader(new FileReader("sample.txt"));
      String line;

      while ((line = reader.readLine()) != null) {
        System.out.println(line);
      }

    } catch (FileNotFoundException e) {
      System.out.println("File not found.");
    } catch (IOException e) {
      System.out.println("Error reading file.");
    } finally {
      try {
        if (reader != null) reader.close();
      } catch (IOException e) {
        System.out.println("Error closing the file.");
      }
```

```
    }
  }
}
```

OUTPUT:

```
D:\OneDrive\Desktop\OOPS\assignment>javac Excp3.java

D:\OneDrive\Desktop\OOPS\assignment>java Excp3
File not found.
```

16D: exception handling 4

CODE:

```
import java.util.Scanner;

public class Excp4 {
  public static void main(String[] args) {
    Scanner sc = new Scanner(System.in);

    try {
      System.out.print("Enter first number: ");
      int a = sc.nextInt();

      System.out.print("Enter second number: ");
      int b = sc.nextInt();

      System.out.println("Choose operation (+, -, *, /): ");
      char op = sc.next().charAt(0);

      double result;

      switch (op) {
        case '+': result = a + b; break;
        case '-': result = a - b; break;
        case '*': result = a * b; break;
        case '/':
          if (b == 0) throw new ArithmeticException("Cannot divide by zero.");
          result = (double) a / b; break;
        default:
          throw new Exception("Invalid operator.");
      }

      System.out.println("Result: " + result);

    } catch (Exception e) {
      System.out.println("Error: " + e.getMessage());
    }
  }
}
```

OUTPUT:

```
D:\OneDrive\Desktop\OOPS\assignment>java Excp4
Enter first number: 10
Enter second number: 10
Choose operation (+, -, *, /):
/
Result: 1.0

D:\OneDrive\Desktop\OOPS\assignment>java Excp4
Enter first number: 10
Enter second number: 0
Choose operation (+, -, *, /):
/
Error: Cannot divide by zero.
```

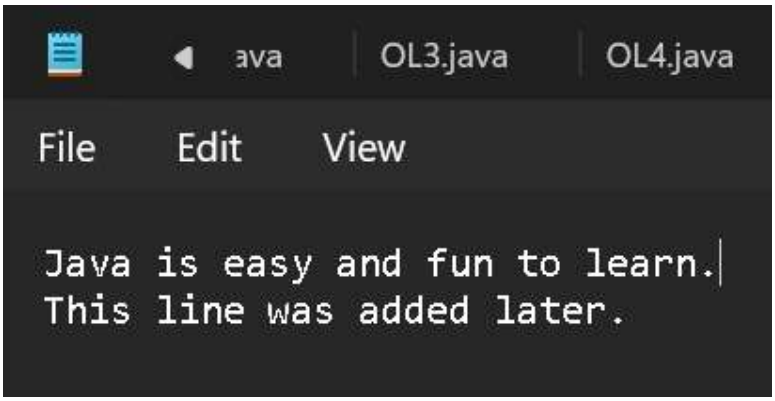
17A: file handling 1

CODE:

```
import java.io.FileWriter;
import java.io.IOException;

public class File1 {
   public static void main(String[] args) {
      try {
         FileWriter writer = new FileWriter("sample.txt", true);
         writer.write("\nThis line was added later.");
         writer.close();
         System.out.println("Appended successfully.");
      } catch (IOException e) {
         System.out.println("An error occurred.");
         e.printStackTrace();
      }
   }
}
```

OUTPUT:

```
D:\OneDrive\Desktop\OOPS\assignment\files>java File1
Appended successfully.

D:\OneDrive\Desktop\OOPS\assignment\files>
```



17B: file handling 2

CODE:

```
import java.io.*;
import java.util.*;

public class File2 {
   static final String FILE_NAME = "sample.txt";

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

      do {
         System.out.println("\n1. Add Person");
         System.out.println("2. View All Person");
         System.out.println("3. Search Person by Name");
         System.out.println("4. Exit");
         System.out.print("Choose: ");
         choice = sc.nextInt();
         sc.nextLine();

         switch (choice) {
            case 1 -> addPerson(sc);
            case 2 -> viewPerson();
            case 3 -> searchPerson(sc);
            case 4 -> System.out.println("Goodbye!");
            default -> System.out.println("Invalid choice.");
         }
      } while (choice != 4);
   }

   static void addPerson(Scanner sc) {
      try (FileWriter fw = new FileWriter(FILE_NAME, true)) {
         System.out.print("Enter name: ");
         String name = sc.nextLine();
         System.out.print("Enter age: ");
         int age = sc.nextInt();
         sc.nextLine();
         fw.write(name + "," + age + "\n");
         System.out.println("Person added.");
      } catch (IOException e) {
         e.printStackTrace();
      }
   }
```

```
  static void viewPerson() {
    try (Scanner fileScanner = new Scanner(new File(FILE_NAME))) {
      System.out.println("All Student Records:");
      while (fileScanner.hasNextLine()) {
        String[] data = fileScanner.nextLine().split(",");
        System.out.println("Name: " + data[0] + ", Age: " + data[1]);
      }
    } catch (FileNotFoundException e) {
      System.out.println("No person records found.");
    }
  }

  static void searchPerson(Scanner sc) {
    System.out.print("Enter name to search: ");
    String searchName = sc.nextLine();
    boolean found = false;

    try (Scanner fileScanner = new Scanner(new File(FILE_NAME))) {
      while (fileScanner.hasNextLine()) {
        String line = fileScanner.nextLine();
        String[] data = line.split(",");
        if (data[0].equalsIgnoreCase(searchName)) {
          System.out.println("Found: Name: " + data[0] + ", Age: " + data[1]);
          found = true;
        }
      }
    } catch (FileNotFoundException e) {
      System.out.println("File not found.");
    }

    if (!found) {
      System.out.println("Person not found.");
    }
  }
}
```

OUTPUT:

```
D:\OneDrive\Desktop\OOPS\assignment\files>java File2

1. Add Person
2. View All Person
3. Search Person by Name
4. Exit
Choose: 1
Enter name: Deepak
Enter age: 18
Person added.

1. Add Person
2. View All Person
3. Search Person by Name
4. Exit
Choose: 3
Enter name to search: Deepak
Found: Name: Deepak, Age: 18

1. Add Person
2. View All Person
3. Search Person by Name
4. Exit
Choose: 4
Goodbye!

D:\OneDrive\Desktop\OOPS\assignment\files>
```

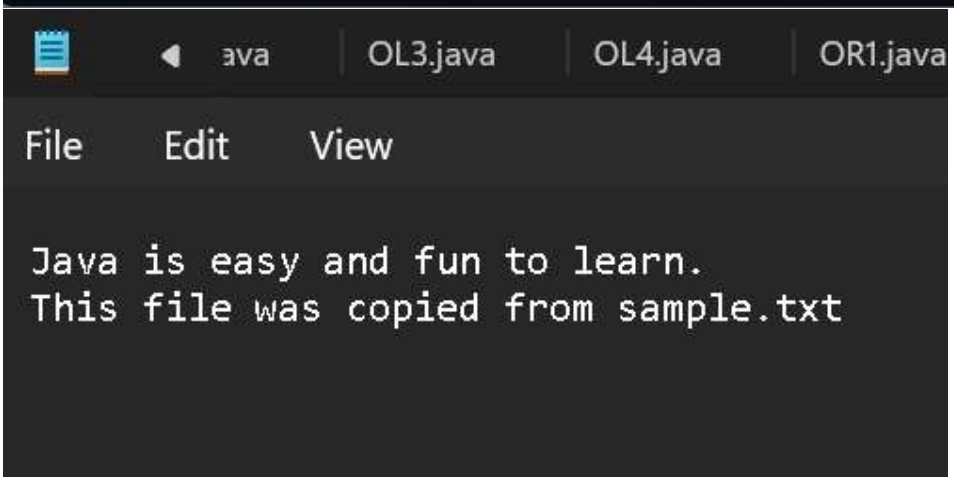
17C: file handling 3

CODE:

```
import java.io.*;

public class File3 {
   public static void main(String[] args) {
      try (BufferedReader reader = new BufferedReader(new FileReader("sample.txt"));
          BufferedWriter writer = new BufferedWriter(new FileWriter("destination.txt"))) {

         String line;
         while ((line = reader.readLine()) != null) {
            writer.write(line);
            writer.newLine();
         }

         System.out.println("File copied successfully.");

      } catch (IOException e) {
         e.printStackTrace();
      }
   }
}
```

OUTPUT:

```
D:\OneDrive\Desktop\OOPS\assignment\files>javac File3.java

D:\OneDrive\Desktop\OOPS\assignment\files>java File3
File copied successfully.

D:\OneDrive\Desktop\OOPS\assignment\files>
```



17D: file handling 4

CODE:

```
import java.io.*;
import java.util.*;

public class File4 {
   static final String FILE_NAME = "sample.txt";

   public static void main(String[] args) {
      Scanner sc = new Scanner(System.in);
      System.out.print("Enter product name: ");
      String name = sc.nextLine();
      System.out.print("Enter price: ");
      double price = sc.nextDouble();
      System.out.print("Enter quantity: ");
      int quantity = sc.nextInt();

      try (FileWriter fw = new FileWriter(FILE_NAME, true)) {
         fw.write(name + "," + price + "," + quantity + "\n");
         System.out.println("Product saved.");
      } catch (IOException e) {
         e.printStackTrace();
      }
   }
}
```

OUTPUT:

```
D:\OneDrive\Desktop\OOPS\assignment\files>java File4
Enter product name: Laptop
Enter price: 80000
Enter quantity: 1
Product saved.

D:\OneDrive\Desktop\OOPS\assignment\files>
```