



**SCHOOL OF
COMPUTING**

LAB RECORD
OBJECT ORIENTED PROGRAMMING
(23CSE111)

NAME: LEENA K

ROLL NO: CH.SC.U4CSE24122

COURSE: CSE-CT

SECTION: B



SCHOOL OF
COMPUTING

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.U4CSE24122 – LEENA K** 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 :08/04/2025

Internal Examiner 1

Internal Examiner 2

INDEX

S.NO	TITLE	PAGE.NO
	UML DIAGRAM	
1.	ATM SYSTEM	
	1.a) Use Case Diagram	6-8
	1.b) Class Diagram	6
	1.c) Sequence Diagram	7
	1.d) Object Diagram	7
	1.e) Deployment Diagram	8
2.	LIBRARY MANAGEMENT SYSTEM	
	2.a) Use Case Diagram	9
	2.b) Class Diagram	9
	2.c) Sequence Diagram	10
	2.d) Object Diagram	10
	2.e) Deployment Diagram	11
3.	BASIC JAVA PROGRAMS	
	3.a) Check Number Nature	12
	3.b) Even or Odd	13
	3.c) Factorial	14
	3.d) Fibonacci Series	15
	3.e) Leap Year	16
	3.f) Multiplication Table	17
	3.g) Palindrome No	18
	3.h) Prime No	19
	3.i) Sum of Digits	20
	3.j) Sum of Natural Numbers	21
	INHERITANCE	
4.	SINGLE INHERITANCE PROGRAMS	
	4.a) Bank System	22-24

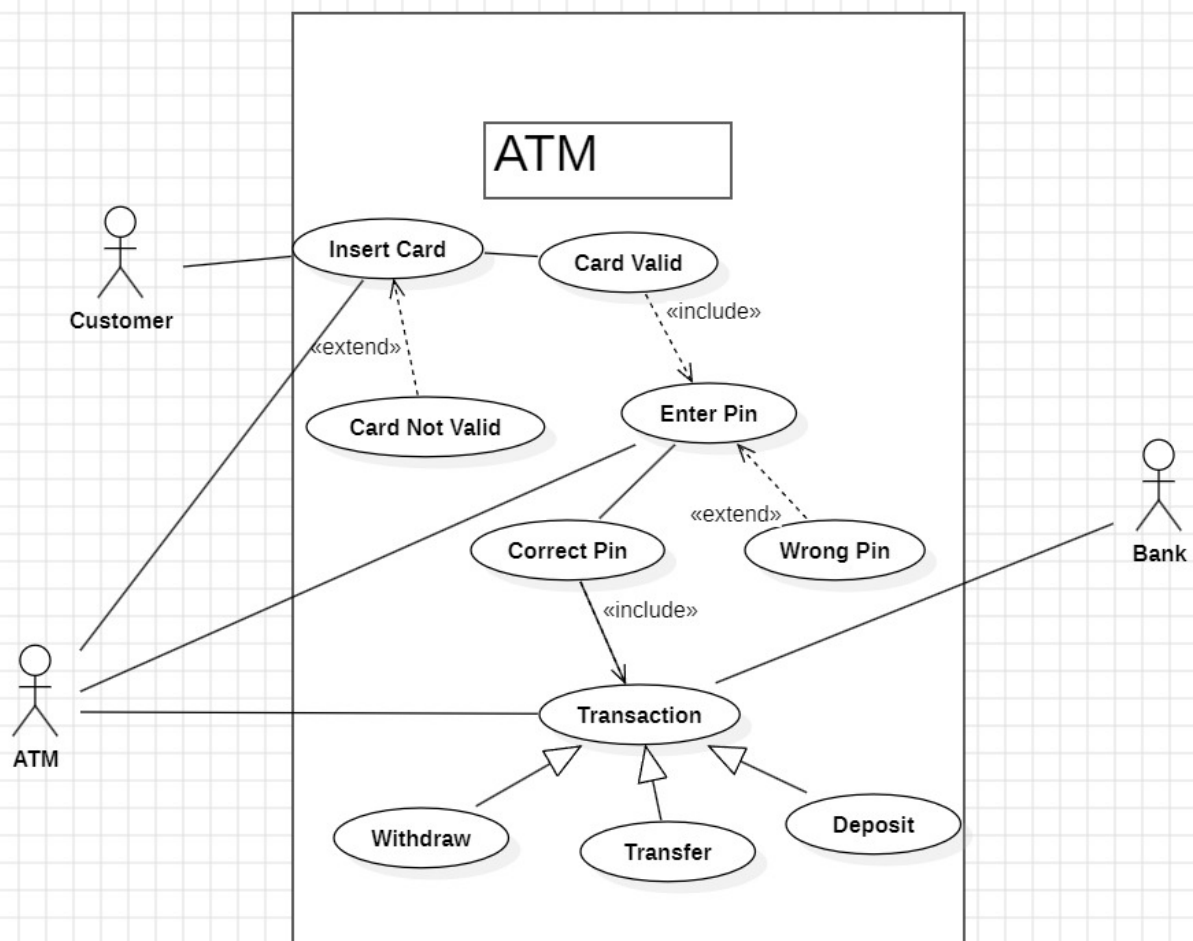
	4.b)Employee Management	24-26
5.	MULTILEVEL INHERITANCE PROGRAMS	27-30
	5.a)University System	27-28
	5.b)Vehicle Hierarchy	28-30
6.	HIERARCHICAL INHERITANCE PROGRAMS	31-35
	6.a)Banking System	31-32
	6.b)Grading System	33-35
7.	HYBRID INHERITANCE PROGRAMS	36-39
	7.a)Fitness System	36-37
	7.b)Hospital System	38-39
	POLYMORPHISM	
8.	CONSTRUCTOR PROGRAMS	40
	8.a)Learning System	40
9.	CONSTRUCTOR OVERLOADING PROGRAMS	41-42
	9.a)Climate Control	41-42
10.	METHOD OVERLOADING PROGRAMS	43-45
	10.a)Clock App	43-44
	10.b)Coffee Shop	44-45
11.	METHOD OVERRIDING PROGRAMS	46-48
	11.a)Music App	46-47
	11.b)Home Control	47-48
	ABSTRACTION	
12.	INTERFACE PROGRAMS	49-55
	12.a)Smart Home	49-50
	12.b)Quiz System	51-52
	12.c)My Music Player	52-53
	12.d)Scientific Calculator	54-55
13.	ABSTRACT CLASS PROGRAMS	56-72
	13.a)Food Order Demo	56-60
	13.b)Shape Demo	61-63
	13.c)Rental Demo	63-68
	13.d)Electronics Demo	68-72
	ENCAPSULATION	
14.	ENCAPSULATION PROGRAMS	73-78
	14.a)Health BMI Calcualtor	73-74
	14.b)Movie Rating System	74-74
	14.c)Shopping Cart System	76-77
	14.d)Temperature Converter	77-78
15.	PACKAGES PROGRAMS	79-104
	15.a)Library System	79-86
	15.b)Bank Account	86-95

	15.c)Password Checker	96-99
	15.d)Scientific Calculator	99-104
16.	EXCEPTION HANDLING PROGRAMS	105-111
	16.a)Age Validator	105-106
	16.b)Math Operations	106-108
	16.c)File Reader	108-109
	16.d)Grade Calculator	110-111
17.	FILE HANDLING PROGRAMS	112-116
	17.a)File Reader	112
	17.b)File Writer	113
	17.c)File Deleter	114-115
	17.d)File Checker	115-116

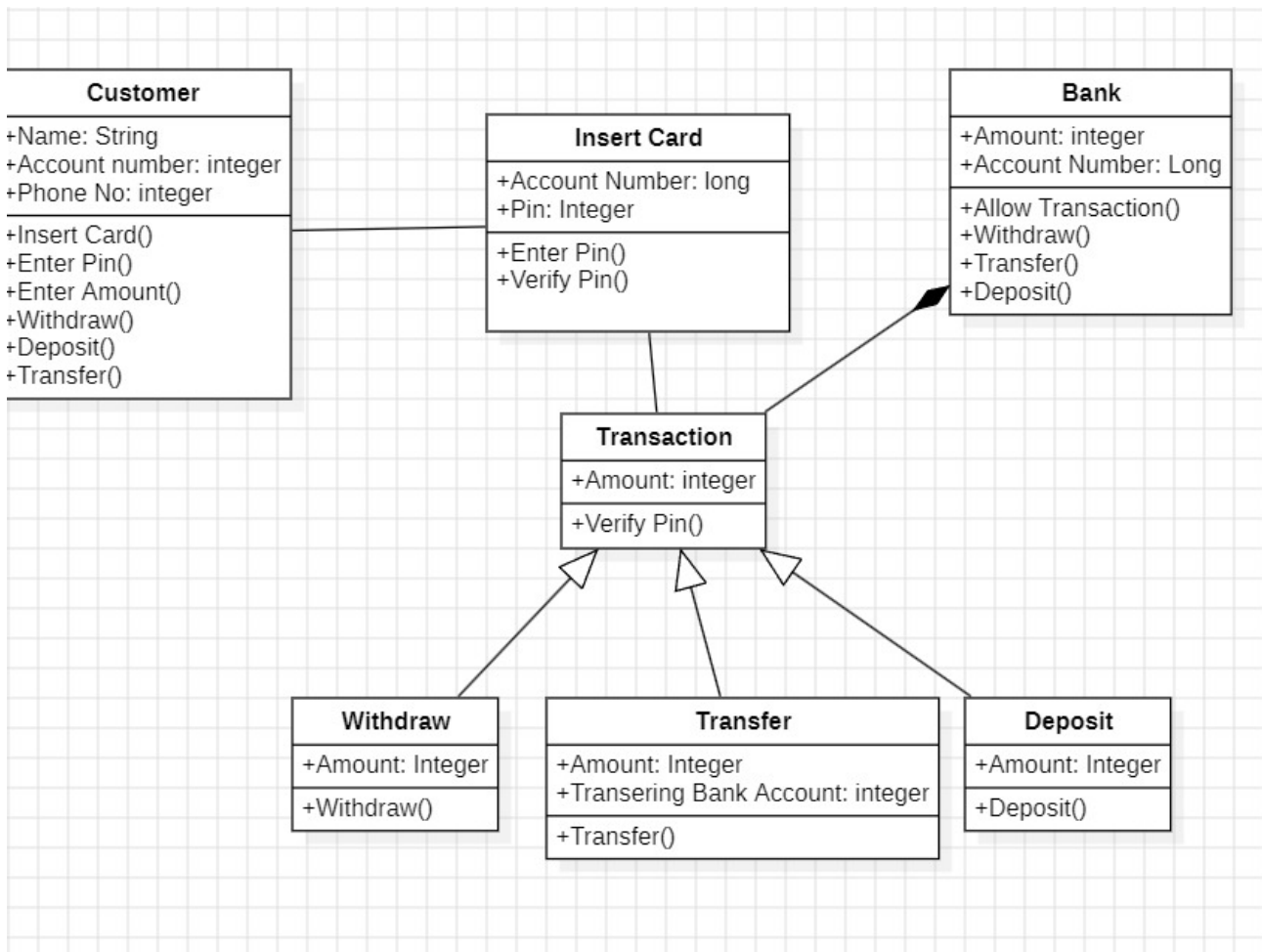
UML DIAGRAMS

1.ATM SYSTEM

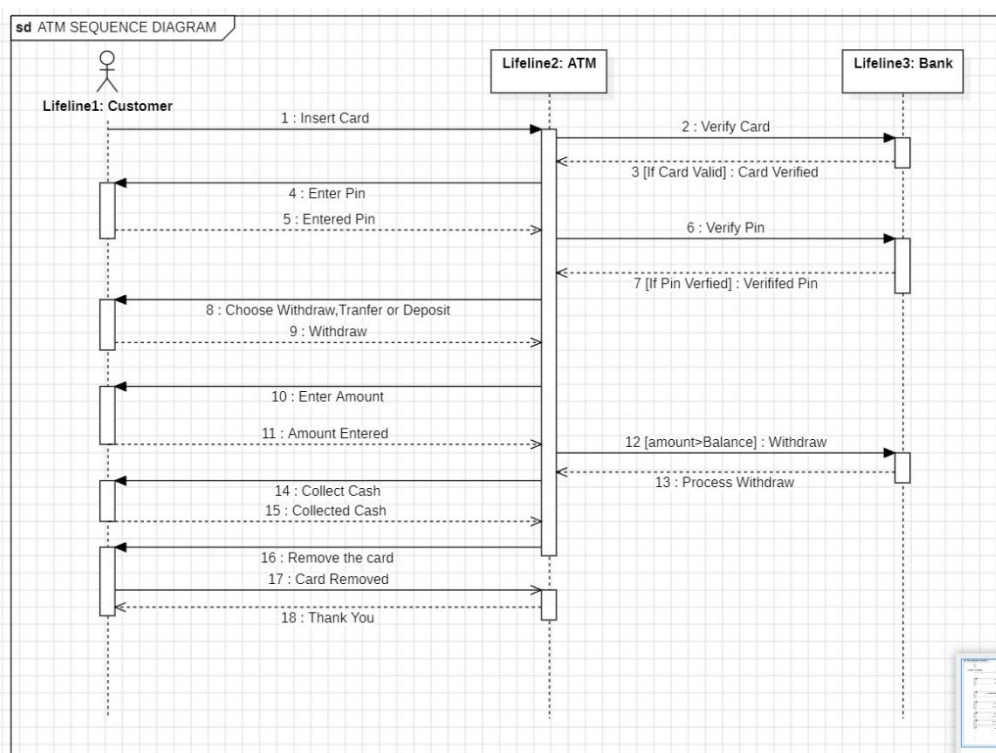
1.a) Use Case Diagram:

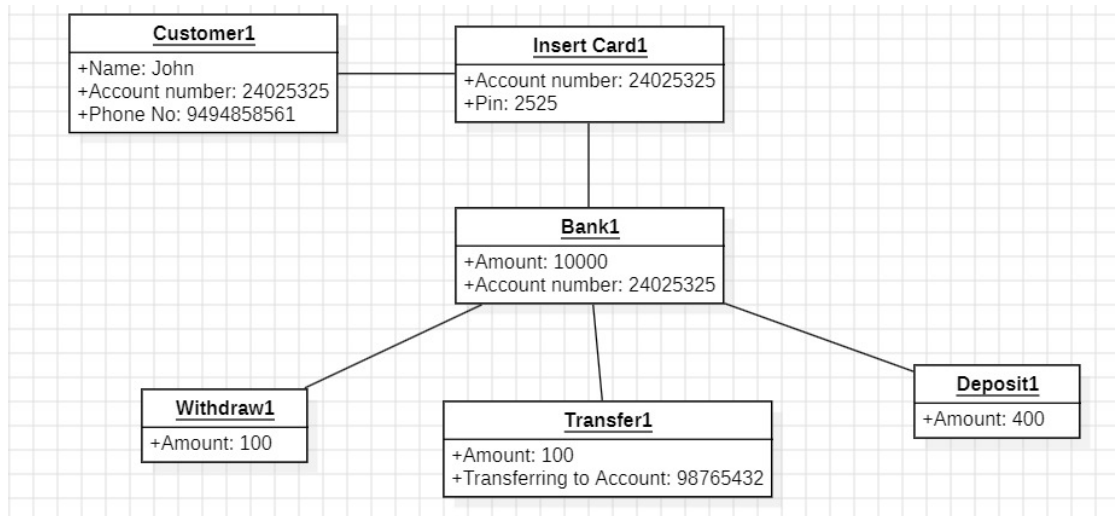
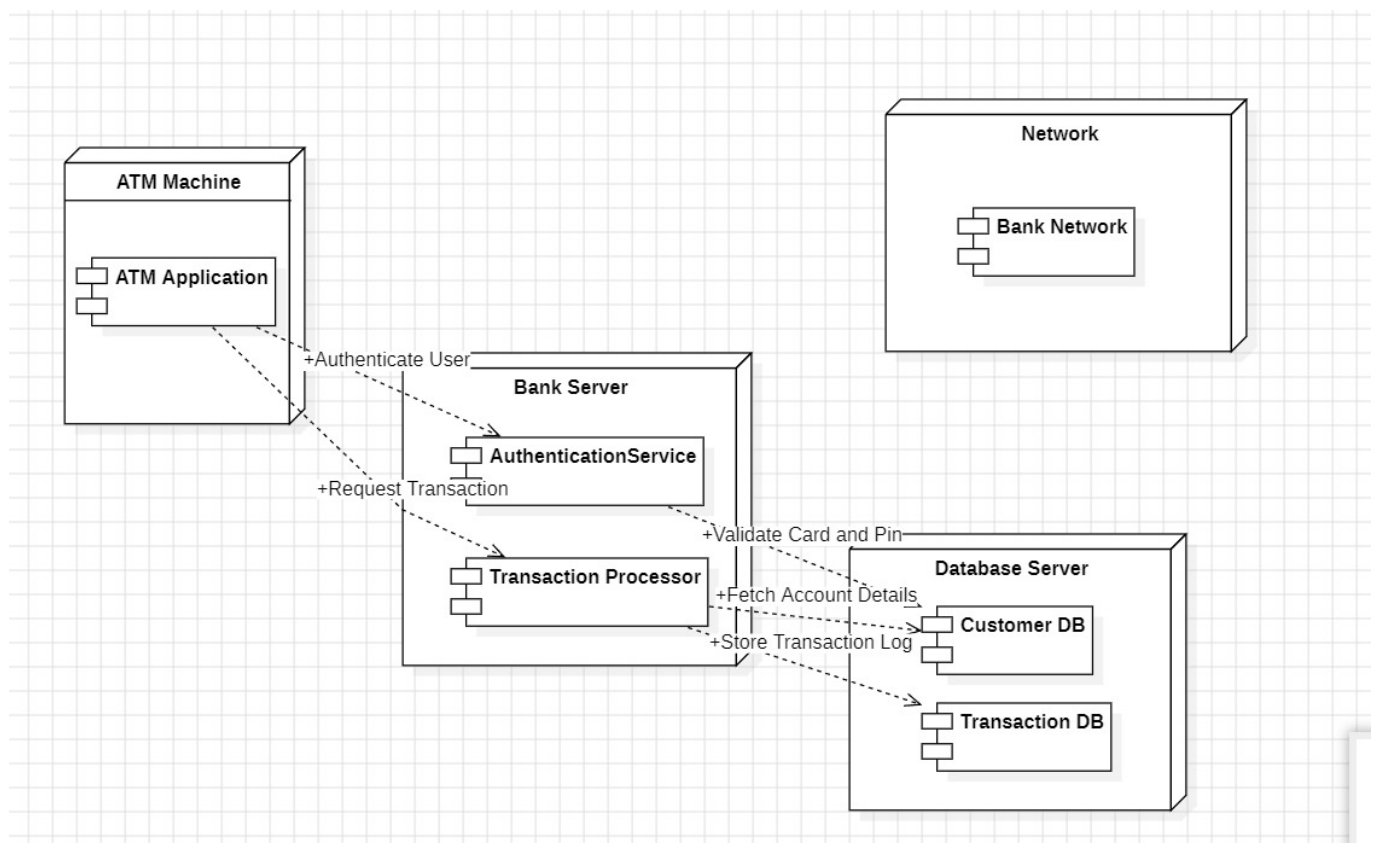


1.b) Class Diagram:



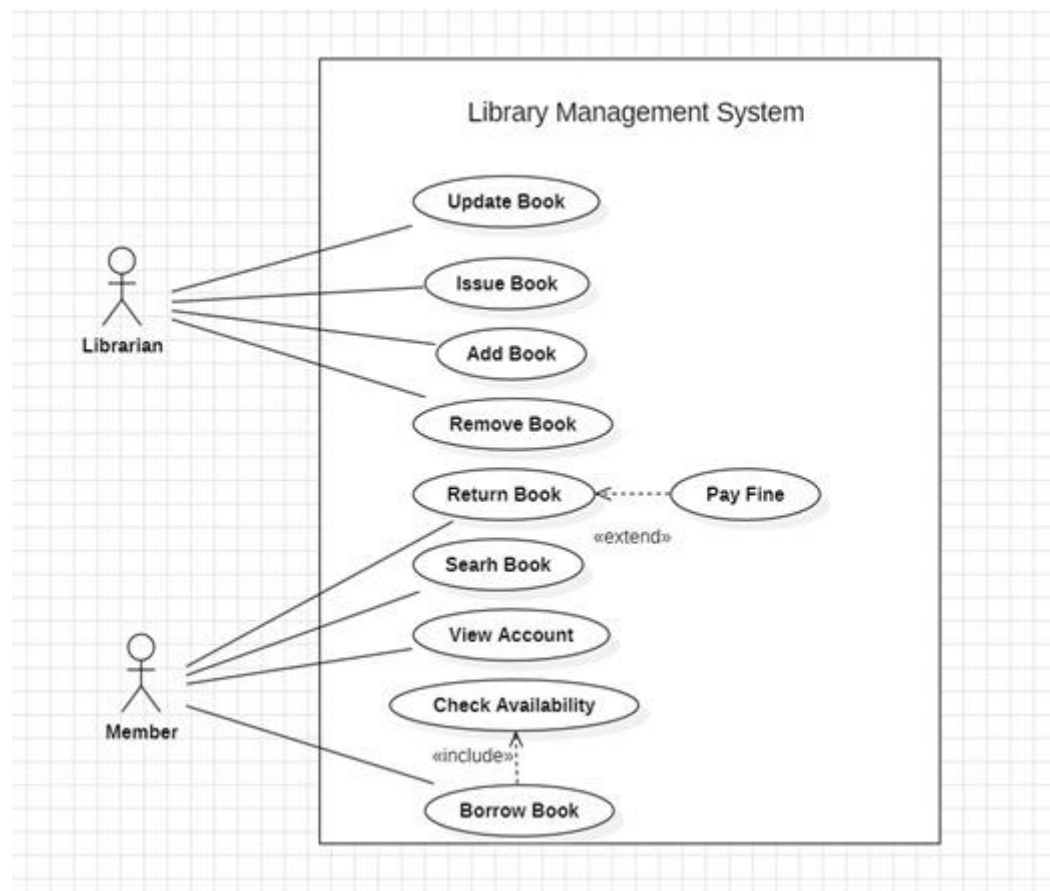
1.c) Sequence Diagram:



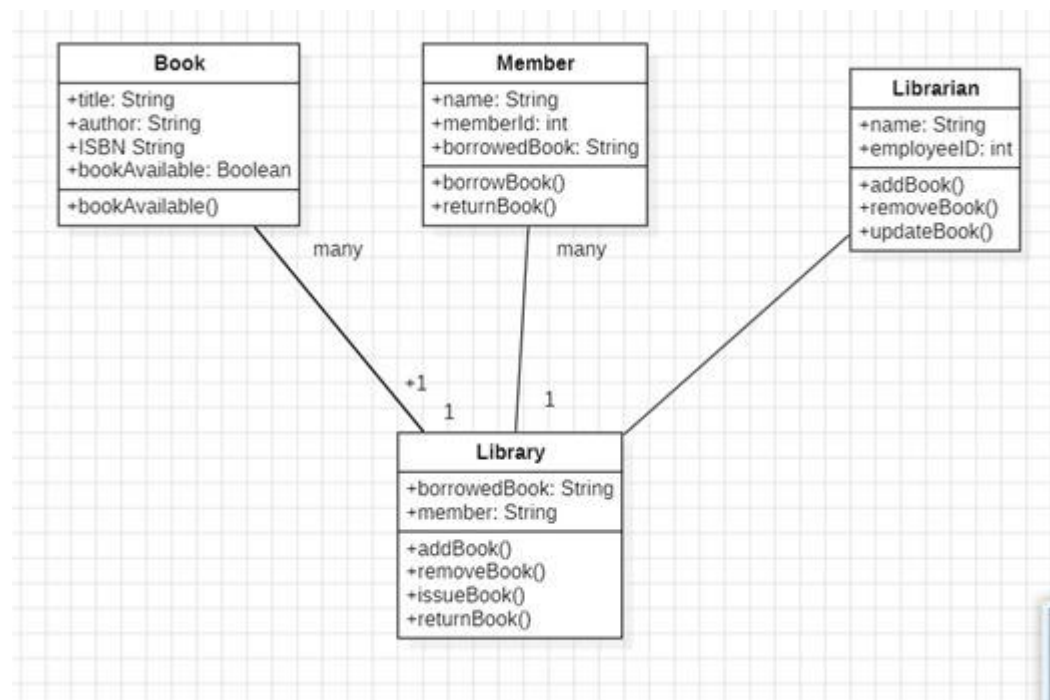
1.d) Object Diagram:**1.e) Deployment Diagram:**

2. LIBRARY MANAGEMENT SYSTEM

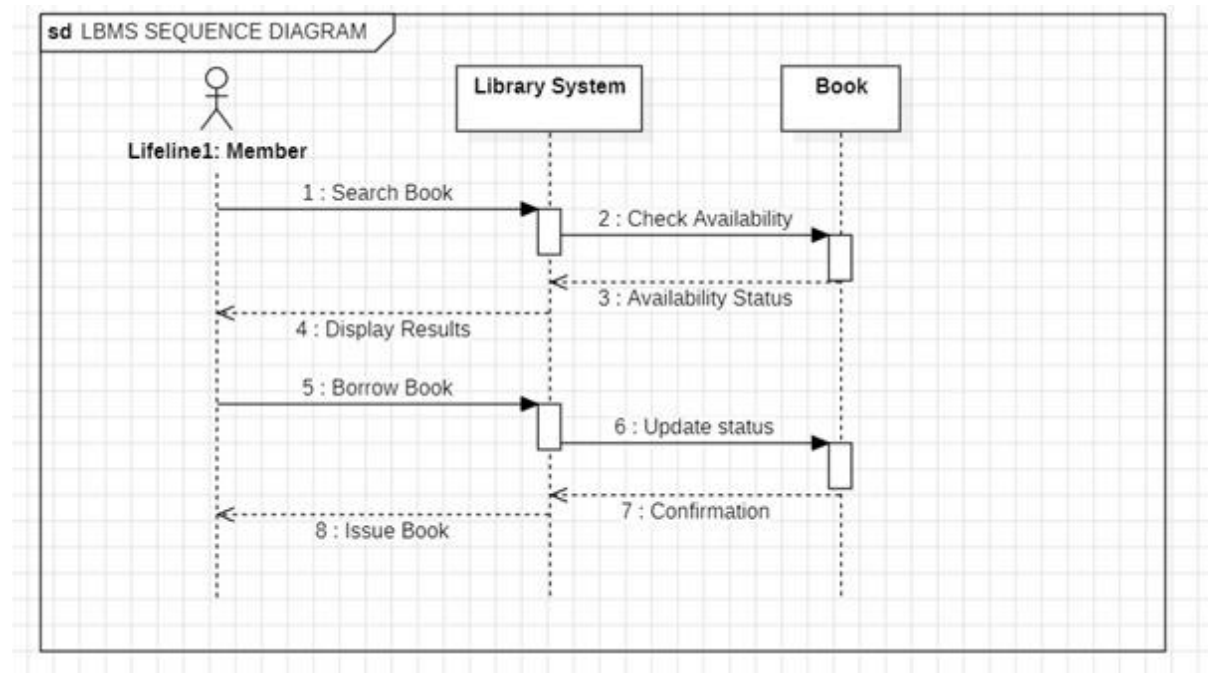
2.a) Use Case Diagram:



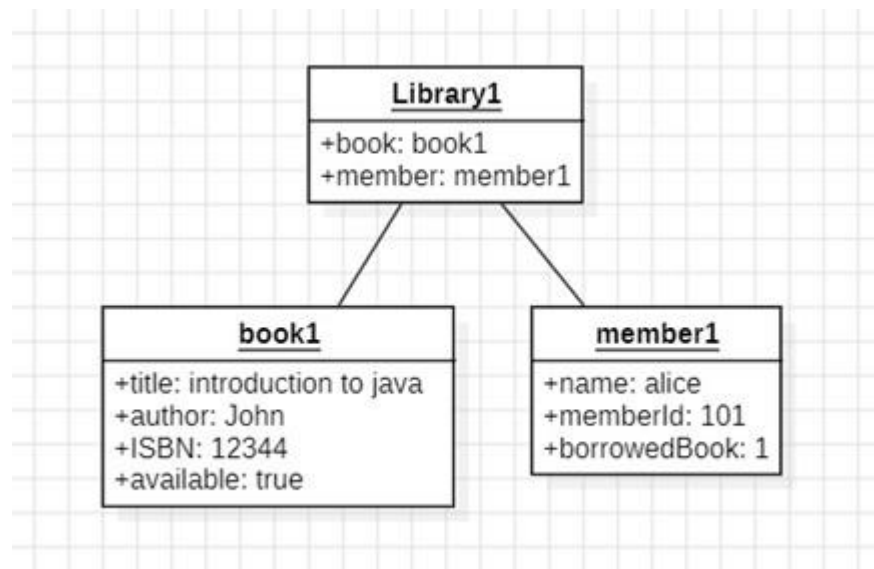
2.b) Class Diagram:

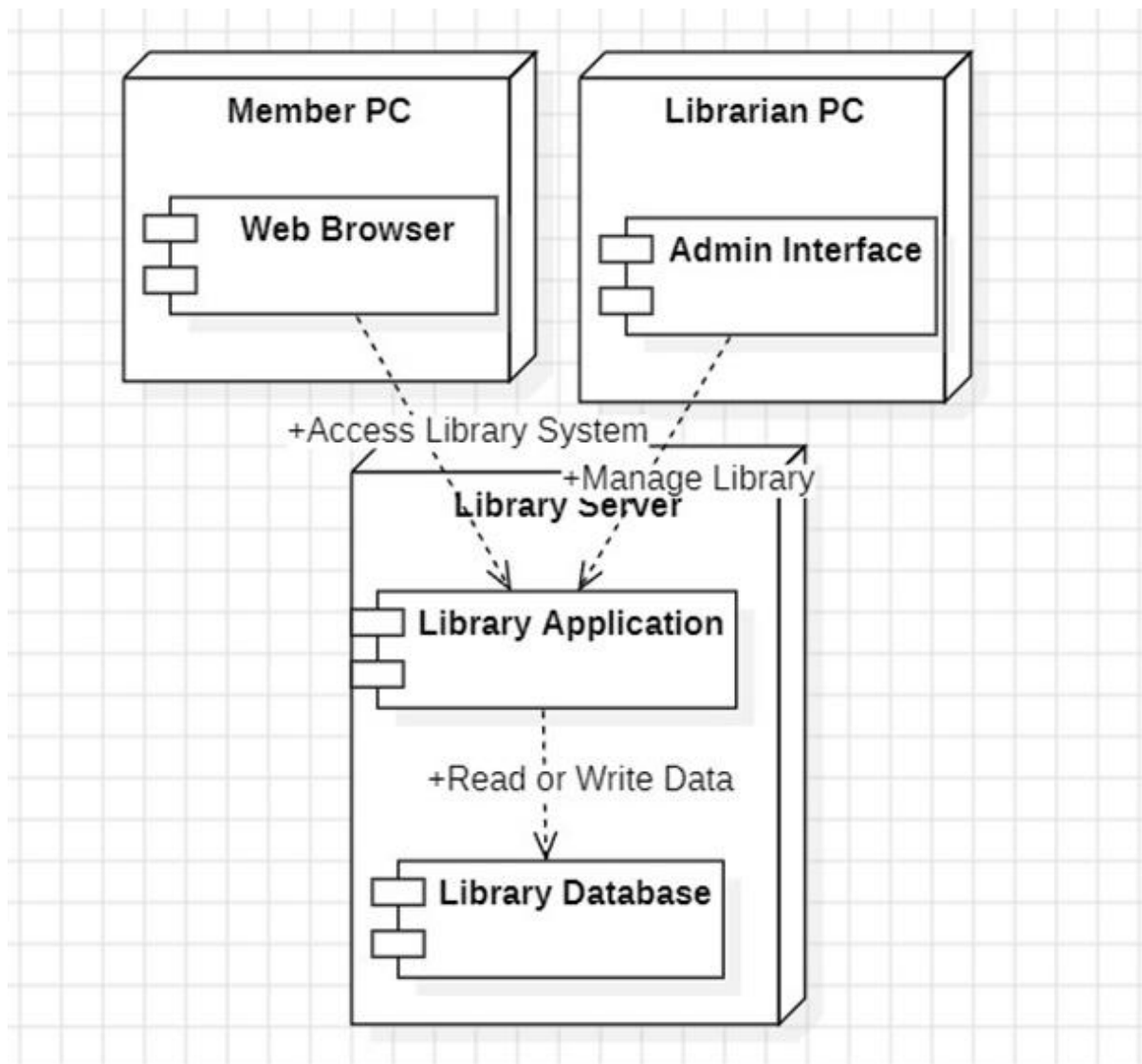


2.c) Sequence Diagram:



2.d) Object Diagram:



2.e)Deployment Diagram:

3.Basic Java Programs

3.a) Check Number Nature:

Code:

```
import java.util.Scanner;

public class CheckNumber {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter a number: ");
        int num = scanner.nextInt();
        if (num > 0) {
            System.out.println(num + " is positive.");
        } else if (num < 0) {
            System.out.println(num + " is negative.");
        } else {
            System.out.println(num + " is zero.");
        }
        scanner.close();
    }
}
```

Output:

```
C:\Users\rohit\Desktop\Java>java CheckNumber
Enter a number: -25
-25 is negative.
```

3.b) Even or Odd:

Code:

```
import java.util.Scanner;

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

        if (num % 2 == 0) {
            System.out.println(num + " is Even");
        } else {
            System.out.println(num + " is Odd");
        }
        scanner.close();
    }
}
```

Output:

```
C:\Users\rohit\Desktop\Java>javac EvenOdd.java

C:\Users\rohit\Desktop\Java>java EvenOdd
Enter a number: 2
2 is Even

C:\Users\rohit\Desktop\Java>|
```

3.c) 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 number: ");
        int num = scanner.nextInt();
        int factorial = 1;
        for (int i = 1; i <= num; i++) {
            factorial *= i;
        }
        System.out.println("Factorial of " + num + " is: " + factorial);
        scanner.close();
    }
}
```

Output:

```
C:\Users\rohit\Desktop\Java>javac Factorial.java

C:\Users\rohit\Desktop\Java>java Factorial
Enter a number: 4
Factorial of 4 is: 24
```

3.d) Fibonacci Series:

Code:

```
import java.util.Scanner;

public class Fibonacci{
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter the number of terms: ");
        int n = scanner.nextInt();
        int firstTerm = 0, secondTerm = 1;
        System.out.println("Fibonacci Series:");
        for (int i = 1; i <= n; i++) {
            System.out.print(firstTerm + " ");
            int nextTerm = firstTerm + secondTerm;
            firstTerm = secondTerm;
            secondTerm = nextTerm;
        }
        scanner.close();
    }
}
```

Output;

```
C:\Users\rohit\Desktop\Java>java Fibonacci
Enter the number of terms: 3
Fibonacci Series:
0 1 1
```


3.e) Leap Year:

Code:

```
import java.util.Scanner;

public class LeapYear {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter a year: ");
        int year = scanner.nextInt();
        if ((year % 4 == 0 && year % 100 != 0) || (year % 400 == 0)) {
            System.out.println(year + " is a leap year.");
        } else {
            System.out.println(year + " is not a leap year.");
        }
        scanner.close();
    }
}
```

Output:

```
C:\Users\rohit\Desktop\Java>javac LeapYear.java

C:\Users\rohit\Desktop\Java>java LeapYear
Enter a year: 2016
2016 is a leap year.
```


3.f) Multiplication Table:

Code:

```
import java.util.Scanner;

public class MultiplicationTable {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter a number: ");
        int num = scanner.nextInt();
        for (int i = 1; i <= 10; i++) {
            System.out.println(num + " x " + i + " = " + (num * i));
        }
        scanner.close();
    }
}
```

Output:

```
C:\Users\rohit\Desktop\Java>javac MultiplicationTable.java

C:\Users\rohit\Desktop\Java>java MultiplicationTable
Enter a number: 5
5 x 1 = 5
5 x 2 = 10
5 x 3 = 15
5 x 4 = 20
5 x 5 = 25
5 x 6 = 30
5 x 7 = 35
5 x 8 = 40
5 x 9 = 45
5 x 10 = 50
```

3.g) Palindrome No:

Code:

```
import java.util.Scanner;

public class PalindromeNo{
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter a number: ");
        int num = scanner.nextInt();
        int reversed = 0, original = num;
        while (num != 0) {
            int digit = num % 10;
            reversed = reversed * 10 + digit;
            num /= 10;
        }
        if (original == reversed) {
            System.out.println(original + " is a palindrome.");
        } else {
            System.out.println(original + " is not a palindrome.");
        }
        scanner.close();
    }
}
```

Output:

```
C:\Users\rohit\Desktop\Java>javac PalindromeNo.java

C:\Users\rohit\Desktop\Java>java PalindromeNo
Enter a number: 5
5 is a palindrome.
```

3.h) Prime No:

Code:

```
import java.util.Scanner;

public class PrimeNo{
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter a number: ");
        int num = scanner.nextInt();
        boolean isPrime = true;
        for (int i = 2; i <= num / 2; i++) {
            if (num % i == 0) {
                isPrime = false;
                break;
            }
        }
        if (isPrime) {
            System.out.println(num + " is a prime number.");
        } else {
            System.out.println(num + " is not a prime number.");
        }
        scanner.close();
    }
}
```

Output:

```
C:\Users\rohit\Desktop\Java>javac PrimeNo.java

C:\Users\rohit\Desktop\Java>java PrimeNo
Enter a number: 6
6 is not a prime number.
```

3.i) Sum of Digits:

Code:

```
import java.util.Scanner;

public class SumOfDigits{
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter a number: ");
        int num = scanner.nextInt();
        int sum = 0;
        while (num != 0) {
            sum += num % 10;
            num /= 10;
        }
        System.out.println("Sum of digits: " + sum);
        scanner.close();
    }
}
```

Output:

```
C:\Users\rohit\Desktop\Java>javac SumOfDigits.java

C:\Users\rohit\Desktop\Java>java SumOfDigits
Enter a number: 61
Sum of digits: 7
```

3.j) Sum of Natural Number:

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 number: ");
        int n = scanner.nextInt();
        int sum = 0;
        for (int i = 1; i <= n; i++) {
            sum += i;
        }
        System.out.println("Sum of first " + n + " natural numbers is: " + sum);
        scanner.close();
    }
}
```

Output:

```
C:\Users\rohit\Desktop\Java>javac SumOfNaturalNumbers.java

C:\Users\rohit\Desktop\Java>java SumOfNaturalNumbers
Enter a number: 4
Sum of first 4 natural numbers is: 10
```

INHERITANCE

4.SINGLE INHERITANCE PROGRAMS

4.a)Bank System

Code:

```
import java.util.Scanner;

class Account {
    String accHolderName;
    long accNumber;
    void getAccountDetails() {
        Scanner obj=new Scanner(System.in);
        System.out.print("Enter Account Holder Name: ");
        accHolderName=obj.nextLine();
        System.out.print("Enter Account Number: ");
        accNumber=obj.nextLong();
    }
    void displayAccountDetails() {
        System.out.println("Account Holder: "+accHolderName);
        System.out.println("Account Number: "+accNumber);
    }
}

class SavingsAccount extends Account{
    double balance, interest;
```

```
void getBalance(){
    Scanner obj=new Scanner(System.in);
    System.out.print("Enter Balance: ");
    balance=obj.nextDouble();
}
void calculateInterest() {
    interest=balance * 0.05;
    System.out.println("Interest Earned: "+interest);
    System.out.println("Total Balance: "+(balance + interest));
}
}
public class BankSystem {
    public static void main(String[] args) {
        SavingsAccount acc = new SavingsAccount();
        acc.getAccountDetails();
        acc.getBalance();
        acc.displayAccountDetails();
        acc.calculateInterest();
    }
}
```

Output:

```
C:\Users\kpriy\OneDrive - Amrita Vishwa Vidyapeetham- Chennai Campus\Java\Lab Manual\Inheritance\Single Inheritance>java c BankSystem.java

C:\Users\kpriy\OneDrive - Amrita Vishwa Vidyapeetham- Chennai Campus\Java\Lab Manual\Inheritance\Single Inheritance>java BankSystem
Enter Account Holder Name: Leena
Enter Account Number: 1234
Enter Balance: 9000
Account Holder: Leena
Account Number: 1234
Interest Earned: 450.0
Total Balance: 9450.0

C:\Users\kpriy\OneDrive - Amrita Vishwa Vidyapeetham- Chennai Campus\Java\Lab Manual\Inheritance\Single Inheritance>
```

4.b)Employee Management

Code:

```
import java.util.Scanner;

class Employee{
    String name;
    int empld;
    void getDetails() {
        Scanner obj= new Scanner(System.in);
        System.out.print("Enter Employee Name: ");
        name=obj.nextLine();
        System.out.print("Enter Employee ID: ");
        empld=obj.nextInt();
    }
    void displayDetails() {
        System.out.println("Employee Name:"+ name);
        System.out.println("Employee ID:"+ empld);
    }
}

class Salary extends Employee{
```



```
double basicPay, hra, da, grossSalary;
void calculateSalary() {
    Scanner obj=new Scanner(System.in);
    System.out.print("Enter Basic Pay: ");
    basicPay=obj.nextDouble();
    hra=0.2*basicPay;
    da=0.1*basicPay;
    grossSalary=basicPay+hra+da;
}
void displaySalary() {
    System.out.println("Basic Pay: "+basicPay);
    System.out.println("HRA: "+hra);
    System.out.println("DA: "+da);
    System.out.println("Gross Salary: "+grossSalary);
}
}
public class EmployeeManagement {
    public static void main(String[] args) {
        Salary emp = new Salary();
        emp.getDetails();
        emp.calculateSalary();
        emp.displayDetails();
        emp.displaySalary();
    }
}
```

}

Output:

```

C:\Users\kpriy\OneDrive - Amrita Vishwa Vidyapeetham- Chennai Campus\Java\Lab Manual\Inheritance\Single Inheritance>jav
c EmployeeManagement.java

C:\Users\kpriy\OneDrive - Amrita Vishwa Vidyapeetham- Chennai Campus\Java\Lab Manual\Inheritance\Single Inheritance>jav
EmployeeManagement
Enter Employee Name: Leena
Enter Employee ID: 1234
Enter Basic Pay: 700
Employee Name:Leena
Employee ID:1234
Basic Pay: 700.0
HRA: 140.0
DA: 70.0
Gross Salary: 910.0

C:\Users\kpriy\OneDrive - Amrita Vishwa Vidyapeetham- Chennai Campus\Java\Lab Manual\Inheritance\Single Inheritance>|

```

5.MULTILEVEL INHERITANCE PROGRAMS

5.a)University System

Code:

```

import java.util.Scanner;

class Person {
    String name;
    void getName() {
        Scanner obj = new Scanner(System.in);
        System.out.print("Enter Name: ");
        name = obj.nextLine();
    }
    void displayName() {
        System.out.println("Name: " + name);
    }
}

class Student extends Person {

```

```
int rollNo;

void getRollNo() {
    Scanner obj = new Scanner(System.in);
    System.out.print("Enter Roll No: ");
    rollNo = obj.nextInt();
}

void displayRollNo() {
    System.out.println("Roll No: " + rollNo);
}
}

class ResearchStudent extends Student {
    String thesisTopic;
    void getThesisTopic() {
        Scanner obj = new Scanner(System.in);
        System.out.print("Enter Thesis Topic: ");
        thesisTopic = obj.nextLine();
    }
    void displayThesisTopic() {
        System.out.println("Thesis Topic: " + thesisTopic);
    }
}

public class UniversitySystem {
    public static void main(String[] args) {
        ResearchStudent student = new ResearchStudent();
    }
}
```

```
        student.getName();  
        student.getRollNo();  
        student.getThesisTopic();  
        student.displayName();  
        student.displayRollNo();  
        student.displayThesisTopic();  
    }  
}
```

Output:

```
C:\Users\kpriy\OneDrive - Amrita Vishwa Vidyapeetham- Chennai Campus\Java\Lab Manual\Inheritance\MultiLevel Inheritancej  
ava UniversitySystem.java  
C:\Users\kpriy\OneDrive - Amrita Vishwa Vidyapeetham- Chennai Campus\Java\Lab Manual\Inheritance\MultiLevel Inheritancej  
ava UniversitySystem  
Enter Name: Leena  
Enter Roll No: 1234  
Enter Thesis Topic: Tree  
Name: Leena  
Roll No: 1234  
Thesis Topic: Tree  
C:\Users\kpriy\OneDrive - Amrita Vishwa Vidyapeetham- Chennai Campus\Java\Lab Manual\Inheritance\MultiLevel Inheritance>
```

5.b)Vehicle Hierarchy

Code:

```
import java.util.Scanner;  
  
class Vehicle {  
    String type;  
    void getType() {  
        Scanner obj = new Scanner(System.in);  
        System.out.print("Enter Vehicle Type: ");  
        type = obj.nextLine();  
    }  
}
```

```
void displayType() {
    System.out.println("Vehicle Type: " + type);
}
}

class Car extends Vehicle {
    String model;
    void getModel() {
        Scanner obj = new Scanner(System.in);
        System.out.print("Enter Car Model: ");
        model = obj.nextLine();
    }
    void displayModel() {
        System.out.println("Car Model: " + model);
    }
}

class SportsCar extends Car {
    int topSpeed;
    void getTopSpeed() {
        Scanner obj = new Scanner(System.in);
        System.out.print("Enter Top Speed (mph): ");
        topSpeed = obj.nextInt();
    }
    void displayTopSpeed() {
        System.out.println("Top Speed: " + topSpeed + " mph");
    }
}
```

```
    }  
}  
  
public class VehicleHierarchy {  
    public static void main(String[] args) {  
        SportsCar car = new SportsCar();  
        car.getType();  
        car.getModel();  
        car.getTopSpeed();  
        car.displayType();  
        car.displayModel();  
        car.displayTopSpeed();  
    }  
}
```

Output:

```
C:\Users\kpriy\OneDrive - Amrita Vishwa Vidyapeetham- Chennai Campus\Java\Lab Manual\Inheritance\MultiLevel Inheritance>javac VehicleHierarchy.java  
C:\Users\kpriy\OneDrive - Amrita Vishwa Vidyapeetham- Chennai Campus\Java\Lab Manual\Inheritance\MultiLevel Inheritance>java VehicleHierarchy  
Enter Vehicle Type: Car  
Enter Car Model: BMW  
Enter Top Speed (mph): 100  
Vehicle Type: Car  
Car Model: BMW  
Top Speed: 100 mph  
C:\Users\kpriy\OneDrive - Amrita Vishwa Vidyapeetham- Chennai Campus\Java\Lab Manual\Inheritance\MultiLevel Inheritance>
```

6.HIERARCHICAL INHERITANCE PROGRAMS

6.a)Banking System

Code:

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

```
class BankAccount {
    String accHolder;
    double balance;
    void getAccountDetails() {
        Scanner obj=new Scanner(System.in);
        System.out.print("Enter account holder name: ");
        accHolder=obj.nextLine();
        System.out.print("Enter balance: ");
        balance=obj.nextDouble();
    }
}

class SavingsAccount extends BankAccount {
    void calculateInterest() {
        System.out.println("Savings Interest: "+(balance*0.05));
    }
}

class CurrentAccount extends BankAccount {
    void calculateMinimumBalance() {
        System.out.println("Minimum Balance: "+(balance*0.1));
    }
}

public class BankingSystem {
    public static void main(String[] args) {
        SavingsAccount sa=new SavingsAccount();
    }
}
```

```

        CurrentAccount ca=new CurrentAccount();
        System.out.println("Savings Account:");
        sa.getAccountDetails();
        sa.calculateInterest();
        System.out.println("Current Account:");
        ca.getAccountDetails();
        ca.calculateMinimumBalance();
    }
}

```

Output:

```

C:\Users\kpriy\OneDrive - Amrita Vishwa Vidyapeetham- Chennai Campus\Java\Lab Manual\Inheritance\Heirarchial Inheritance
>javac BankingSystem.java
C:\Users\kpriy\OneDrive - Amrita Vishwa Vidyapeetham- Chennai Campus\Java\Lab Manual\Inheritance\Heirarchial Inheritance
>java BankingSystem
Savings Account:
Enter account holder name: Leena
Enter balance: 1200
Savings Interest: 60.0
Current Account:
Enter account holder name: Leesh
Enter balance: 1300
Minimum Balance: 130.0
C:\Users\kpriy\OneDrive - Amrita Vishwa Vidyapeetham- Chennai Campus\Java\Lab Manual\Inheritance\Heirarchial Inheritance
>

```

6.b)Grading System

Code:

//Q.University Grading System

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

```
class Student {
```

```
    String name;
```

```
    int rollNo;
```

```
    void getStudentDetails() {
```

```
        Scanner obj=new Scanner(System.in);
```



```
        System.out.print("Enter student name: ");
        name=obj.nextLine();
        System.out.print("Enter roll number: ");
        rollNo=obj.nextInt();
    }
}

class ScienceStudent extends Student {
    int physicsMarks,chemistryMarks;
    void getScienceMarks() {
        Scanner obj=new Scanner(System.in);
        System.out.print("Enter Physics marks: ");
        physicsMarks=obj.nextInt();
        System.out.print("Enter Chemistry marks: ");
        chemistryMarks=obj.nextInt();
    }
    void displayScienceAverage() {
        System.out.println("Science Average:
"+((physicsMarks+chemistryMarks)/2.0));
    }
}

class ArtsStudent extends Student {
    int historyMarks,economicsMarks;
    void getArtsMarks() {
        Scanner obj=new Scanner(System.in);
```

```
        System.out.print("Enter History marks: ");
        historyMarks=obj.nextInt();

        System.out.print("Enter Economics marks: ");
        economicsMarks=obj.nextInt();
    }

    void displayArtsAverage() {
        System.out.println("Arts Average:
"+((historyMarks+economicsMarks)/2.0));
    }
}

public class GradingSystem {
    public static void main(String[] args) {
        ScienceStudent sci=new ScienceStudent();
        ArtsStudent art=new ArtsStudent();
        System.out.println("Science Student:");
        sci.getStudentDetails();
        sci.getScienceMarks();
        sci.displayScienceAverage();
        System.out.println("Arts Student:");
        art.getStudentDetails();
        art.getArtsMarks();
        art.displayArtsAverage();
    }
}
```

Output:

```
C:\Users\kpriy\OneDrive - Amrita Vishwa Vidyapeetham- Chennai Campus\Java\Lab Manual\Inheritance\Heirarchial Inheritance
>javac GradingSystem.java

C:\Users\kpriy\OneDrive - Amrita Vishwa Vidyapeetham- Chennai Campus\Java\Lab Manual\Inheritance\Heirarchial Inheritance
>java GradingSystem
Science Student:
Enter student name: Leena
Enter roll number: 90
Enter Physics marks: 100
Enter Chemistry marks: 97
Science Average: 98.5
Arts Student:
Enter student name: Leesh
Enter roll number: 98
Enter History marks: 98
Enter Economics marks: 99
Arts Average: 98.5

C:\Users\kpriy\OneDrive - Amrita Vishwa Vidyapeetham- Chennai Campus\Java\Lab Manual\Inheritance\Heirarchial Inheritance
>|
```

7. HYBRID INHERITANCE PROGRAMS

7.a)Fitness System

Code:

```
import java.util.Scanner;

class FitnessDevice {
    String deviceName;

    FitnessDevice() {
        Scanner obj=new Scanner(System.in);
        System.out.print("Enter device name: ");
        deviceName=obj.nextLine();
    }
}

class Wearable extends FitnessDevice {
    boolean waterproof;

    Wearable() {
```

```
        super();
        Scanner obj=new Scanner(System.in);
        System.out.print("Is it waterproof? (true/false): ");
        waterproof=obj.nextBoolean();
    }
}

class GymEquipment extends FitnessDevice {
    int weightCapacity;
    GymEquipment() {
        super();
        Scanner obj=new Scanner(System.in);
        System.out.print("Enter weight capacity (kg): ");
        weightCapacity=obj.nextInt();
    }
}

class SmartBand extends Wearable {
    String healthMetrics;
    SmartBand() {
        super();
        Scanner obj=new Scanner(System.in);
        System.out.print("Enter health metrics tracked: ");
        healthMetrics=obj.nextLine();
    }
}
```

```

public class FitnessSystem {
    public static void main(String[] args) {
        SmartBand band=new SmartBand();
        GymEquipment treadmill=new GymEquipment();
    }
}

```

Output:

```

C:\Users\kpriy\OneDrive - Amrita Vishwa Vidyapeetham- Chennai Campus\Java\Lab Manual\Inheritance\Hybrid Inheritance>java
c FitnessSystem.java

C:\Users\kpriy\OneDrive - Amrita Vishwa Vidyapeetham- Chennai Campus\Java\Lab Manual\Inheritance\Hybrid Inheritance>java
FitnessSystem
Enter device name: Samsung
Is it waterproof? (true/false): true
Enter health metrics tracked: 5
Enter device name: iPhone
Enter weight capacity (kg): 2

C:\Users\kpriy\OneDrive - Amrita Vishwa Vidyapeetham- Chennai Campus\Java\Lab Manual\Inheritance\Hybrid Inheritance>|

```

7.b)Hospital System

Code:

```

import java.util.Scanner;

class HospitalStaff {
    String staffId;

    HospitalStaff() {
        Scanner obj=new Scanner(System.in);
        System.out.print("Enter staff ID: ");
        staffId=obj.nextLine();
    }
}

class Doctor extends HospitalStaff {

```

```
String specialization;

Doctor() {
    super();
    Scanner obj=new Scanner(System.in);
    System.out.print("Enter specialization: ");
    specialization=obj.nextLine();
}

class Nurse extends HospitalStaff {
    String shift;
    Nurse() {
        super();
        Scanner obj=new Scanner(System.in);
        System.out.print("Enter shift (day/night): ");
        shift=obj.nextLine();
    }
}

class Surgeon extends Doctor {
    String surgeryType;
    Surgeon() {
        super();
        Scanner obj=new Scanner(System.in);
        System.out.print("Enter surgery type: ");
        surgeryType=obj.nextLine();
    }
}
```

```

    }
}

public class HospitalSystem {
    public static void main(String[] args) {
        Surgeon s1=new Surgeon();
        Nurse n1=new Nurse();
    }
}

```

Output:

```

C:\Users\kpriy\OneDrive - Amrita Vishwa Vidyapeetham- Chennai Campus\Java\Lab Manual\Inheritance\Hybrid Inheritance>java
c HospitalSystem.java
C:\Users\kpriy\OneDrive - Amrita Vishwa Vidyapeetham- Chennai Campus\Java\Lab Manual\Inheritance\Hybrid Inheritance>java
HospitalSystem
Enter staff ID: 12
Enter specialization: Cardiology
Enter surgery type: Heart
Enter staff ID: 13
Enter shift (day/night): night
C:\Users\kpriy\OneDrive - Amrita Vishwa Vidyapeetham- Chennai Campus\Java\Lab Manual\Inheritance\Hybrid Inheritance>|

```

POLYMORPHISM

8.CONSTRUCTOR PROGRAMS

8.a)Learning System

Code:

```

import java.util.Scanner;

class Course {
    String courseCode;
    int maxEnrollment;
    Course() {
        Scanner obj=new Scanner(System.in);
    }
}

```

```
        System.out.print("Enter course code: ");
        courseCode=obj.nextLine();
        System.out.print("Enter max enrollment: ");
        maxEnrollment=obj.nextInt();
    }
    void checkAvailability(int currentEnroll) {
        System.out.println("Available seats: "+(maxEnrollment-
currentEnroll));
    }
}

public class LearningSystem {
    public static void main(String[] args) {
        Course javaCourse=new Course();
        javaCourse.checkAvailability(15);
    }
}
```

Output:

```
C:\Users\kpriy\OneDrive - Amrita Vishwa Vidyapeetham- Chennai Campus\Java\Lab Manual\Polymorphism\Constructor Program>javac LearningSystem.java
C:\Users\kpriy\OneDrive - Amrita Vishwa Vidyapeetham- Chennai Campus\Java\Lab Manual\Polymorphism\Constructor Program>java LearningSystem
Enter course code: 12
Enter max enrollment: 34
Available seats: 19
C:\Users\kpriy\OneDrive - Amrita Vishwa Vidyapeetham- Chennai Campus\Java\Lab Manual\Polymorphism\Constructor Program>
```

9.CONSTRUCTOR OVERLOADING PROGRAMS

9.a)Climate Control

Code:


```
import java.util.Scanner;

class Thermostat {
    String deviceId;
    double currentTemp;
    double targetTemp;

    Thermostat() {
        Scanner obj=new Scanner(System.in);
        System.out.print("Enter device ID: ");
        deviceId=obj.nextLine();
        System.out.print("Enter current temperature: ");
        currentTemp=obj.nextDouble();
        targetTemp=22.0;
    }

    Thermostat(String id, double temp) {
        deviceId=id;
        currentTemp=temp;
        targetTemp=22.0;
    }

    Thermostat(String id, double currTemp, double targTemp) {
        deviceId=id;
        currentTemp=currTemp;
        targetTemp=targTemp;
    }
}
```

```
void adjustTemperature() {  
    System.out.println("Adjusting from "+currentTemp+"°C to  
"+targetTemp+"°C");  
}  
}  
  
public class ClimateControl {  
    public static void main(String[] args) {  
        Thermostat t1=new Thermostat();  
        Thermostat t2=new Thermostat("LivingRoom",25.5);  
        Thermostat t3=new Thermostat("Bedroom",18.0,20.0);  
        t1.adjustTemperature();  
        t2.adjustTemperature();  
        t3.adjustTemperature();  
    }  
}
```

Output:

```
C:\Users\kpriy\OneDrive - Amrita Vishwa Vidyapeetham- Chennai Campus\Java\Lab Manual\Polymorphism\Constructor Overloadin  
g Program>javac ClimateControl.java  
C:\Users\kpriy\OneDrive - Amrita Vishwa Vidyapeetham- Chennai Campus\Java\Lab Manual\Polymorphism\Constructor Overloadin  
g Program>java ClimateControl  
Enter device ID: 12  
Enter current temperature: 36  
Adjusting from 36.0°C to 22.0°C  
Adjusting from 25.5°C to 22.0°C  
Adjusting from 18.0°C to 20.0°C  
C:\Users\kpriy\OneDrive - Amrita Vishwa Vidyapeetham- Chennai Campus\Java\Lab Manual\Polymorphism\Constructor Overloadin  
g Program>
```

10.METHOD OVERLOADING PROGRAMS

10.a)Clock App

Code:

```
import java.util.Scanner;

class AlarmClock {

    Scanner obj=new Scanner(System.in);

    void setAlarm() {

        System.out.print("Enter hour: ");

        int hour=obj.nextInt();

        System.out.println("Alarm set for "+hour+":00");

    }

    void setAlarm(int hour) {

        System.out.print("Enter minutes: ");

        int minute=obj.nextInt();

        System.out.println("Alarm set for "+hour+": "+minute);

    }

    void setAlarm(int hour,int minute) {

        obj.nextLine(); // Clear buffer

        System.out.print("Enter sound: ");

        String sound=obj.nextLine();

        System.out.println("Alarm set for "+hour+": "+minute+" with "+sound+" sound");

    }

}

public class ClockApp {

    public static void main(String[] args) {

        AlarmClock myClock=new AlarmClock();
```

```
        myClock.setAlarm();  
        myClock.setAlarm(7);  
        myClock.setAlarm(7,30);  
    }  
}
```

Output:

```
C:\Users\kpriy\OneDrive - Amrita Vishwa Vidyapeetham- Chennai Campus\Java\Lab Manual\Polymorphism\Method Overloading Program>javac ClockApp.java  
C:\Users\kpriy\OneDrive - Amrita Vishwa Vidyapeetham- Chennai Campus\Java\Lab Manual\Polymorphism\Method Overloading Program>java ClockApp  
Enter hour: 24  
Alarm set for 24:00  
Enter minutes: 30  
Alarm set for 7:30  
Enter sound: birds  
Alarm set for 7:30 with birds sound  
C:\Users\kpriy\OneDrive - Amrita Vishwa Vidyapeetham- Chennai Campus\Java\Lab Manual\Polymorphism\Method Overloading Program>
```

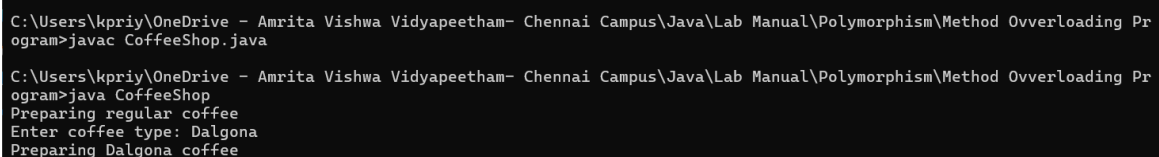
10.b)Coffee Shop

Code:

```
import java.util.Scanner;  
  
class CoffeeOrder {  
    Scanner obj=new Scanner(System.in);  
    void prepareCoffee() {  
        System.out.println("Preparing regular coffee");  
    }  
    void prepareCoffee(String type) {  
        System.out.print("Enter coffee type: ");  
        type=obj.nextLine();  
        System.out.println("Preparing "+type+" coffee");  
    }  
}
```

```
void prepareCoffee(String type,int sugars) {  
    System.out.print("Enter coffee type: ");  
    type=obj.nextLine();  
    System.out.print("Enter sugar amount: ");  
    sugars=obj.nextInt();  
    System.out.println("Preparing "+type+" coffee with "+sugars+"  
sugars");  
}  
}  
  
public class CoffeeShop {  
    public static void main(String[] args) {  
        CoffeeOrder order=new CoffeeOrder();  
        order.prepareCoffee();  
        order.prepareCoffee("");  
        order.obj.nextLine(); // Clear buffer  
        order.prepareCoffee("",0);  
    }  
}
```

Output:



```
C:\Users\kpriy\OneDrive - Amrita Vishwa Vidyapeetham- Chennai Campus\Java\Lab Manual\Polymorphism\Method Overloading Program>javac CoffeeShop.java  
C:\Users\kpriy\OneDrive - Amrita Vishwa Vidyapeetham- Chennai Campus\Java\Lab Manual\Polymorphism\Method Overloading Program>java CoffeeShop  
Preparing regular coffee  
Enter coffee type: Dalgona  
Preparing Dalgona coffee
```

11.METHOD OVERRIDING PROGRAMS

11.a)Music App

Code:

```
import java.util.Scanner;

class AudioPlayer {
    void play() {
        System.out.println("Playing audio at normal quality");
    }
}

class MP3Player extends AudioPlayer {
    void play() {
        System.out.println("Playing MP3 with enhanced compression");
    }
}

public class MusicApp {
    public static void main(String[] args) {
        Scanner obj = new Scanner(System.in);
        System.out.print("Enter player type (1=Audio, 2=MP3): ");
        int choice = obj.nextInt();

        AudioPlayer player = (choice == 2) ? new MP3Player() : new
AudioPlayer();

        player.play();
    }
}
```

Output:

```
C:\Users\kpriy\OneDrive - Amrita Vishwa Vidyapeetham- Chennai Campus\Java\Lab Manual\Polymorphism\Method Overriding Program>javac MusicApp.java

C:\Users\kpriy\OneDrive - Amrita Vishwa Vidyapeetham- Chennai Campus\Java\Lab Manual\Polymorphism\Method Overriding Program>java MusicApp
Enter player type (1=Audio, 2=MP3): 2
Playing MP3 with enhanced compression

C:\Users\kpriy\OneDrive - Amrita Vishwa Vidyapeetham- Chennai Campus\Java\Lab Manual\Polymorphism\Method Overriding Program>
```

11.b)Home Control

Code:

```
import java.util.Scanner;

class Appliance {
    void powerOn() {
        System.out.println("Appliance turned on normally");
    }
}

class SmartAppliance extends Appliance {
    void powerOn() {
        System.out.println("Appliance turned on with voice command");
    }
}

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

        System.out.print("Use smart control? (y/n): ");

        String choice = obj.next();

        Appliance device = choice.equalsIgnoreCase("y") ? new SmartAppliance() : new Appliance();
    }
}
```

```

        device.powerOn();
    }
}

```

Output:

```

C:\Users\kpriy\OneDrive - Amrita Vishwa Vidyapeetham- Chennai Campus\Java\Lab Manual\Polymorphism\Method Overriding Program>javac HomeControl.java
C:\Users\kpriy\OneDrive - Amrita Vishwa Vidyapeetham- Chennai Campus\Java\Lab Manual\Polymorphism\Method Overriding Program>java HomeControl
Use smart control? (y/n): y
Appliance turned on with voice command
C:\Users\kpriy\OneDrive - Amrita Vishwa Vidyapeetham- Chennai Campus\Java\Lab Manual\Polymorphism\Method Overriding Program>

```

ABSTRACTION

12.INTERFACE PROGRAMS

12.a)Smart Home

Code:

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

```

interface SmartDevice {
    void turnOn();
    void turnOff();
    void adjust(int level);
}

```

```

class Light implements SmartDevice {
    public void turnOn() { System.out.println("Light turned ON"); }
    public void turnOff() { System.out.println("Light turned OFF"); }
    public void adjust(int brightness) { System.out.println("Brightness set to: " + brightness); }
}

```



```
}
```

```
class Fan implements SmartDevice {  
    public void turnOn() { System.out.println("Fan turned ON"); }  
    public void turnOff() { System.out.println("Fan turned OFF"); }  
    public void adjust(int speed) { System.out.println("Fan speed set  
to: " + speed); }  
}
```

```
public class SmartHome {  
    public static void main(String[] args) {  
        Scanner sc = new Scanner(System.in);  
        System.out.println("Choose device (1-Light, 2-Fan): ");  
        int choice = sc.nextInt();  
  
        SmartDevice device;  
        if (choice == 1) device = new Light();  
        else device = new Fan();  
  
        System.out.println("1-Turn On, 2-Turn Off, 3-Adjust: ");  
        int action = sc.nextInt();  
  
        if (action == 1) device.turnOn();  
        else if (action == 2) device.turnOff();
```

```
        else {  
            System.out.println("Enter adjustment level: ");  
            int level = sc.nextInt();  
            device.adjust(level);  
        }  
        sc.close();  
    }  
}
```

Output:

```
C:\Users\kpriy\OneDrive - Amrita Vishwa Vidyapeetham- Chennai Campus\Java\Lab Manual\Abstraction\Interface Program>javac  
SmartHome.java  
C:\Users\kpriy\OneDrive - Amrita Vishwa Vidyapeetham- Chennai Campus\Java\Lab Manual\Abstraction\Interface Program>java  
SmartHome  
Choose device (1-Light, 2-Fan):  
1  
1-Turn On, 2-Turn Off, 3-Adjust:  
3  
Enter adjustment level:  
45  
Brightness set to: 45  
C:\Users\kpriy\OneDrive - Amrita Vishwa Vidyapeetham- Chennai Campus\Java\Lab Manual\Abstraction\Interface Program>
```

12.b) Quiz System

Code:

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

```
interface QuizQuestion {  
    void displayQuestion();  
    boolean checkAnswer(String userAnswer);  
}
```

```
class MCQ implements QuizQuestion {  
    private String question;
```

```
private String correctOption;

MCQ(String q, String correct) {
    question = q;
    correctOption = correct;
}

public void displayQuestion() { System.out.println(question); }
public boolean checkAnswer(String userAnswer) { return
userAnswer.equalsIgnoreCase(correctOption); }
}

public class QuizSystem {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        QuizQuestion q = new MCQ("What is 2+2? (A)3 (B)4 (C)5", "B");

        q.displayQuestion();
        System.out.println("Your answer (A/B/C): ");
        String ans = sc.next();

        System.out.println(q.checkAnswer(ans) ? "Correct!" : "Wrong!");
        sc.close();
    }
}
```

```
}
```

Output:

```
C:\Users\kpriy\OneDrive - Amrita Vishwa Vidyapeetham- Chennai Campus\Java\Lab Manual\Abstraction\Interface Program>javac QuizSystem.java
C:\Users\kpriy\OneDrive - Amrita Vishwa Vidyapeetham- Chennai Campus\Java\Lab Manual\Abstraction\Interface Program>java QuizSystem
What is 2+2? (A)3 (B)4 (C)5
Your answer (A/B/C):
B
Correct!
C:\Users\kpriy\OneDrive - Amrita Vishwa Vidyapeetham- Chennai Campus\Java\Lab Manual\Abstraction\Interface Program>
```

12.c)My Music Player

Code:

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

```
interface MusicPlayer {
```

```
    void play();
```

```
    void pause();
```

```
    void next();
```

```
}
```

```
class BasicPlayer implements MusicPlayer {
```

```
    public void play() { System.out.println("Playing music..."); }
```

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

```
    public void next() { System.out.println("Next track."); }
```

```
}
```

```
public class MyMusicPlayer {
```

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

```
Scanner sc = new Scanner(System.in);
MusicPlayer player = new BasicPlayer();

System.out.println("1-Play, 2-Pause, 3-Next: ");
int choice = sc.nextInt();

switch(choice) {
    case 1: player.play(); break;
    case 2: player.pause(); break;
    case 3: player.next(); break;
    default: System.out.println("Invalid choice!");
}

sc.close();
}
```

Output:

```
C:\Users\kpriy\OneDrive - Amrita Vishwa Vidyapeetham- Chennai Campus\Java\Lab Manual\Abstraction\Interface Program>javac
MyMusicPlayer.java
C:\Users\kpriy\OneDrive - Amrita Vishwa Vidyapeetham- Chennai Campus\Java\Lab Manual\Abstraction\Interface Program>java
MyMusicPlayer
1-Play, 2-Pause, 3-Next:
1
Playing music...
```

12.d)Scientific Calculator

Code:

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

```
interface Calculator {
```

```
double add(double a, double b);  
double subtract(double a, double b);  
double multiply(double a, double b);  
double divide(double a, double b);  
}
```

```
class BasicCalculator implements Calculator {  
    public double add(double a, double b) { return a + b; }  
    public double subtract(double a, double b) { return a - b; }  
    public double multiply(double a, double b) { return a * b; }  
    public double divide(double a, double b) { return (b != 0) ? a / b :  
Double.NaN; }  
}
```

```
public class ScientificCalc {  
    public static void main(String[] args) {  
        Scanner sc = new Scanner(System.in);  
        Calculator calc = new BasicCalculator();  
  
        System.out.println("Enter two numbers: ");  
        double x = sc.nextDouble();  
        double y = sc.nextDouble();  
  
        System.out.println("1-Add, 2-Subtract, 3-Multiply, 4-Divide: ");
```

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

switch(op) {
    case 1: System.out.println("Result: " + calc.add(x, y)); break;
    case 2: System.out.println("Result: " + calc.subtract(x, y));
break;
    case 3: System.out.println("Result: " + calc.multiply(x, y));
break;
    case 4: System.out.println("Result: " + calc.divide(x, y)); break;
    default: System.out.println("Invalid choice!");
}
sc.close();
}
}
```

Output:

```
C:\Users\kpriy\OneDrive - Amrita Vishwa Vidyapeetham- Chennai Campus\Java\Lab Manual\Abstraction\Interface Program>javac
ScientificCalc.java
C:\Users\kpriy\OneDrive - Amrita Vishwa Vidyapeetham- Chennai Campus\Java\Lab Manual\Abstraction\Interface Program>java
ScientificCalc
Enter two numbers:
2
2
1-Add, 2-Subtract, 3-Multiply, 4-Divide:
1
Result: 4.0
```

13. ABSTRACT CLASS PROGRAMS

13.a)Food Order Demo

Code:

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

```
abstract class FoodItem {
```

```
protected String name;  
protected double price;
```

```
FoodItem(String name, double price) {  
    this.name = name;  
    this.price = price;  
}
```

```
abstract void prepare();  
abstract void displayDetails();  
}
```

```
class Pizza extends FoodItem {  
    private String size;  
    private String[] toppings;
```

```
Pizza(String name, double price, String size, String[] toppings) {  
    super(name, price);  
    this.size = size;  
    this.toppings = toppings;  
}
```

```
void prepare() {
```



```
        System.out.println("Preparing " + size + " pizza with " +  
toppings.length + " toppings");  
    }
```

```
void displayDetails() {  
    System.out.println("Pizza Details:");  
    System.out.println("Name: " + name);  
    System.out.println("Size: " + size);  
    System.out.println("Price: $" + price);  
    System.out.print("Toppings: ");  
    for (String topping : toppings) {  
        System.out.print(topping + " ");  
    }  
    System.out.println();  
}  
}
```

```
class Burger extends FoodItem {  
    private boolean hasCheese;  
    private boolean hasFries;
```

```
    Burger(String name, double price, boolean hasCheese, boolean  
hasFries) {  
        super(name, price);
```

```
        this.hasCheese = hasCheese;
        this.hasFries = hasFries;
    }

    void prepare() {
        System.out.println("Preparing burger" + (hasCheese ? " with
cheese" : "") +
            (hasFries ? " with fries" : ""));
    }

    void displayDetails() {
        System.out.println("Burger Details:");
        System.out.println("Name: " + name);
        System.out.println("Price: $" + price);
        System.out.println("Has Cheese: " + (hasCheese ? "Yes" : "No"));
        System.out.println("Includes Fries: " + (hasFries ? "Yes" : "No"));
    }
}

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

        System.out.println("Enter food type (pizza/burger):");
```

```
String type = sc.next();

System.out.println("Enter item name:");
String name = sc.next();

System.out.println("Enter price:");
double price = sc.nextDouble();

FoodItem item;

if (type.equalsIgnoreCase("pizza")) {
    System.out.println("Enter size (small/medium/large):");
    String size = sc.next();

    System.out.println("Enter number of toppings:");
    int numToppings = sc.nextInt();
    String[] toppings = new String[numToppings];

    for (int i = 0; i < numToppings; i++) {
        System.out.println("Enter topping " + (i+1) + ":");
        toppings[i] = sc.next();
    }

    item = new Pizza(name, price, size, toppings);
}
```

```
    } else {  
        System.out.println("Include cheese? (true/false):");  
        boolean cheese = sc.nextBoolean();  
        System.out.println("Include fries? (true/false):");  
        boolean fries = sc.nextBoolean();  
  
        item = new Burger(name, price, cheese, fries);  
    }  
  
    item.displayDetails();  
    item.prepare();  
    sc.close();  
}  
}
```

Output:

```
C:\Users\kpriy\OneDrive - Amrita Vishwa Vidyapeetham- Chennai Campus\Java\Lab Manual\Abstraction\Abstract Class Programj  
ava FoodOrderDemo.java  
  
C:\Users\kpriy\OneDrive - Amrita Vishwa Vidyapeetham- Chennai Campus\Java\Lab Manual\Abstraction\Abstract Class Programj  
ava FoodOrderDemo  
Enter food type (pizza/burger):  
pizza  
Enter item name:  
Machroni  
Enter price:  
299  
Enter size (small/medium/large):  
small  
Enter number of toppings:  
3  
Enter topping 1:  
Peppe  
Enter topping 2:  
chilly  
Enter topping 3:  
prawn  
Pizza Details:  
Name: Machroni  
Size: small  
Price: $299.0  
Toppings: Peppe chilly prawn  
Preparing small pizza with 3 toppings
```

13.b)Shape Demo

Code:

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

```
abstract class Shape {  
    abstract double calculateArea();  
}
```

```
class Circle extends Shape {  
    private double radius;
```

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

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

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

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

```
}

double calculateArea() {
    return length * width;
}
}

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

        System.out.println("Enter shape type (circle/rectangle):");
        String shapeType = sc.next();

        Shape shape;

        if (shapeType.equalsIgnoreCase("circle")) {
            System.out.println("Enter radius:");
            double radius = sc.nextDouble();
            shape = new Circle(radius);
        } else {
            System.out.println("Enter length and width:");
            double length = sc.nextDouble();
            double width = sc.nextDouble();
        }
    }
}
```

```
        shape = new Rectangle(length, width);
    }

    System.out.println("Area: " + shape.calculateArea());
    sc.close();
}
}
```

Output:

```
C:\Users\kpriy\OneDrive - Amrita Vishwa Vidyapeetham- Chennai Campus\Java\Lab Manual\Abstraction\Abstract Class Program>javac ShapeDemo.java
C:\Users\kpriy\OneDrive - Amrita Vishwa Vidyapeetham- Chennai Campus\Java\Lab Manual\Abstraction\Abstract Class Program>java ShapeDemo
Enter shape type (circle/rectangle):
circle
Enter radius:
3
Area: 28.274333882308138
```

13.c)Rental Demo

Code:

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

```
abstract class Vehicle {
```

```
    protected String model;
```

```
    protected String licensePlate;
```

```
    Vehicle(String model, String licensePlate) {
```

```
        this.model = model;
```

```
        this.licensePlate = licensePlate;
```

```
    }
```

```
    abstract double calculateRentalCost(int days);
```

```
    abstract void displayDetails();  
}
```

```
class Car extends Vehicle {  
    private int seatingCapacity;  
    private double dailyRate;
```

```
    Car(String model, String licensePlate, int seatingCapacity, double  
dailyRate) {  
        super(model, licensePlate);  
        this.seatingCapacity = seatingCapacity;  
        this.dailyRate = dailyRate;  
    }
```

```
    double calculateRentalCost(int days) {  
        return dailyRate * days;  
    }
```

```
    void displayDetails() {  
        System.out.println("Car Details:");  
        System.out.println("Model: " + model);  
        System.out.println("License Plate: " + licensePlate);  
        System.out.println("Seating Capacity: " + seatingCapacity);  
        System.out.println("Daily Rate: $" + dailyRate);  
    }
```



```
}  
}
```

```
class Bike extends Vehicle {  
    private boolean hasHelmet;  
    private double hourlyRate;
```

```
    Bike(String model, String licensePlate, boolean hasHelmet, double  
    hourlyRate) {
```

```
        super(model, licensePlate);  
        this.hasHelmet = hasHelmet;  
        this.hourlyRate = hourlyRate;  
    }
```

```
    double calculateRentalCost(int hours) {  
        return hourlyRate * hours;  
    }
```

```
    void displayDetails() {  
        System.out.println("Bike Details:");  
        System.out.println("Model: " + model);  
        System.out.println("License Plate: " + licensePlate);  
        System.out.println("Includes Helmet: " + (hasHelmet ? "Yes" :  
        "No"));
```

```
        System.out.println("Hourly Rate: $" + hourlyRate);  
    }  
}
```

```
public class RentalDemo {  
    public static void main(String[] args) {  
        Scanner sc = new Scanner(System.in);  
  
        System.out.println("Enter vehicle type (car/bike:");  
        String type = sc.next();  
  
        System.out.println("Enter model:");  
        String model = sc.next();  
  
        System.out.println("Enter license plate:");  
        String plate = sc.next();  
  
        Vehicle vehicle;  
  
        if (type.equalsIgnoreCase("car")) {  
            System.out.println("Enter seating capacity:");  
            int capacity = sc.nextInt();  
            System.out.println("Enter daily rate:");  
            double rate = sc.nextDouble();  
        }  
    }  
}
```

```
        vehicle = new Car(model, plate, capacity, rate);

        System.out.println("Enter rental days:");
        int days = sc.nextInt();

        System.out.println("Total cost: $" +
vehicle.calculateRentalCost(days));
    } else {
        System.out.println("Does it include helmet? (true/false:");
        boolean helmet = sc.nextBoolean();
        System.out.println("Enter hourly rate:");
        double rate = sc.nextDouble();
        vehicle = new Bike(model, plate, helmet, rate);

        System.out.println("Enter rental hours:");
        int hours = sc.nextInt();

        System.out.println("Total cost: $" +
vehicle.calculateRentalCost(hours));
    }

    vehicle.displayDetails();
    sc.close();
}
}
```

Output:

```
C:\Users\kpriy\OneDrive - Amrita Vishwa Vidyapeetham- Chennai Campus\Java\Lab Manual\Abstraction\Abstract Class Program>javac RentalDemo.java
C:\Users\kpriy\OneDrive - Amrita Vishwa Vidyapeetham- Chennai Campus\Java\Lab Manual\Abstraction\Abstract Class Program>java RentalDemo
Enter vehicle type (car/bike):
car
Enter model:
bmw
Enter license plate:
tn20
Enter seating capacity:
5
Enter daily rate:
203
Enter rental days:
56
Total cost: $11368.0
Car Details:
Model: bmw
License Plate: tn20
Seating Capacity: 5
Daily Rate: $203.0
```

13.d)Electronics Demo

Code:

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

```
abstract class ElectronicDevice {
```

```
    protected String brand;
```

```
    protected String model;
```

```
    protected double price;
```

```
    ElectronicDevice(String brand, String model, double price) {
```

```
        this.brand = brand;
```

```
        this.model = model;
```

```
        this.price = price;
```

```
    }
```

```
    abstract void displayFeatures();
```

```
    abstract double calculateDiscount();
```

```
}
```

```
class Smartphone extends ElectronicDevice {  
    private String os;  
    private int storageGB;  
  
    Smartphone(String brand, String model, double price, String os, int  
storageGB) {  
        super(brand, model, price);  
        this.os = os;  
        this.storageGB = storageGB;  
    }  
  
    void displayFeatures() {  
        System.out.println("Smartphone Features:");  
        System.out.println("Brand: " + brand);  
        System.out.println("Model: " + model);  
        System.out.println("OS: " + os);  
        System.out.println("Storage: " + storageGB + "GB");  
        System.out.println("Price: $" + price);  
    }  
  
    double calculateDiscount() {  
        return price * 0.1; // 10% discount  
    }  
}
```

```
class Laptop extends ElectronicDevice {  
    private String processor;  
    private int ramGB;  
  
    Laptop(String brand, String model, double price, String processor,  
int ramGB) {  
        super(brand, model, price);  
        this.processor = processor;  
        this.ramGB = ramGB;  
    }  
  
    void displayFeatures() {  
        System.out.println("Laptop Features:");  
        System.out.println("Brand: " + brand);  
        System.out.println("Model: " + model);  
        System.out.println("Processor: " + processor);  
        System.out.println("RAM: " + ramGB + "GB");  
        System.out.println("Price: $" + price);  
    }  
  
    double calculateDiscount() {  
        return price * 0.15; // 15% discount  
    }  
}
```

```
}
```

```
public class ElectronicsDemo {  
    public static void main(String[] args) {  
        Scanner sc = new Scanner(System.in);  
  
        System.out.println("Enter device type (smartphone/laptop):");  
        String type = sc.next();  
  
        System.out.println("Enter brand:");  
        String brand = sc.next();  
  
        System.out.println("Enter model:");  
        String model = sc.next();  
  
        System.out.println("Enter price:");  
        double price = sc.nextDouble();  
  
        ElectronicDevice device;  
  
        if (type.equalsIgnoreCase("smartphone")) {  
            System.out.println("Enter OS:");  
            String os = sc.next();  
            System.out.println("Enter storage (GB):");
```

```

        int storage = sc.nextInt();

        device = new Smartphone(brand, model, price, os, storage);
    } else {

        System.out.println("Enter processor:");

        String processor = sc.next();

        System.out.println("Enter RAM (GB):");

        int ram = sc.nextInt();

        device = new Laptop(brand, model, price, processor, ram);
    }

    device.displayFeatures();

    System.out.println("Discount: $" + device.calculateDiscount());

    System.out.println("Final Price: $" + (device.price -
device.calculateDiscount()));

    sc.close();
}
}

```

Output:

```

C:\Users\kpriy\OneDrive - Amrita Vishwa Vidyapeetham- Chennai Campus\Java\Lab Manual\Abstraction\Abstract Class Program>javac ElectronicsDemo.java
C:\Users\kpriy\OneDrive - Amrita Vishwa Vidyapeetham- Chennai Campus\Java\Lab Manual\Abstraction\Abstract Class Program>java ElectronicsDemo
Enter device type (smartphone/laptop):
smartphone
Enter brand:
vivo
Enter model:
2019
Enter price:
233334
Enter OS:
google
Enter storage (GB):
256
Smartphone Features:
Brand: vivo
Model: 2019
OS: google
Storage: 256GB
Price: $233334.0
Discount: $23333.4
Final Price: $210000.6

```

ENCAPSULATION

14.ENCAPSULATION PROGRAMS

14.a)Health BMI Calculator

Code:

```
import java.util.Scanner;

class Person {
    private double weight;
    private double height;
    public void setWeight(double weight) { this.weight = weight; }
    public void setHeight(double height) { this.height = height; }
    public double calculateBMI() { return weight / (height * height); }
    public String getBMICategory() {
        double bmi = calculateBMI();
        if(bmi < 18.5) return "Underweight";
        else if(bmi < 25) return "Normal";
        else if(bmi < 30) return "Overweight";
        else return "Obese";
    }
}

public class Main {
    public static void main(String[] args) {
        Scanner obj = new Scanner(System.in);
        Person person = new Person();
        System.out.print("Enter weight (kg): ");
        person.setWeight(obj.nextDouble());
```

```
        System.out.print("Enter height (m): ");
        person.setHeight(obj.nextDouble());
        System.out.println("BMI: " + person.calculateBMI());
        System.out.println("Category: " + person.getBMICategory());
    }
}
```

Output:

```
C:\Users\kpriy\OneDrive - Amrita Vishwa Vidyapeetham- Chennai Campus\Java\Lab Manual\Encapsulation\Health BMI Calculator
>javac Main.java
C:\Users\kpriy\OneDrive - Amrita Vishwa Vidyapeetham- Chennai Campus\Java\Lab Manual\Encapsulation\Health BMI Calculator
>java Main
Enter weight (kg): 56
Enter height (m): 192
BMI: 0.0015190972222222222
Category: Underweight
```

14.b)Movie Rating System

Code:

```
import java.util.Scanner;

class Movie {
    private String name;
    private double[] ratings;

    public void setName(String name) { this.name = name; }
    public void setRatings(double[] ratings) { this.ratings = ratings; }
    public double calculateAverageRating() {
        double sum = 0;
        for(double rating : ratings) sum += rating;
        return sum / ratings.length;
    }

    public String getName() { return name; }
}
```

```
}  
  
public class Main {  
    public static void main(String[] args) {  
        Scanner obj = new Scanner(System.in);  
        Movie movie = new Movie();  
        System.out.print("Enter movie name: ");  
        movie.setName(obj.nextLine());  
        System.out.print("Enter number of ratings: ");  
        int n = obj.nextInt();  
        double[] ratings = new double[n];  
        System.out.println("Enter ratings:");  
        for(int i=0;i<n;i++) ratings[i] = obj.nextDouble();  
        movie.setRatings(ratings);  
        System.out.println(movie.getName() + "'s average rating: " +  
movie.calculateAverageRating());  
    }  
}
```

Output:

```
C:\Users\kpriy\OneDrive - Amrita Vishwa Vidyapeetham- Chennai Campus\Java\Lab Manual\Encapsulation\Movie Rating System>j  
avac Main.java  
  
C:\Users\kpriy\OneDrive - Amrita Vishwa Vidyapeetham- Chennai Campus\Java\Lab Manual\Encapsulation\Movie Rating System>j  
ava Main  
Enter movie name: RRR  
Enter number of ratings: 4  
Enter ratings:  
2  
2  
4  
1  
RRR's average rating: 2.25
```

14.c)Shopping Cart System

Code:

```
import java.util.Scanner;

class Cart {

    private String[] items;
    private double[] prices;

    public void setItems(String[] items) { this.items = items; }
    public void setPrices(double[] prices) { this.prices = prices; }
    public double calculateTotal() {
        double total = 0;
        for(double price : prices) total += price;
        return total;
    }
}

public class Main {
    public static void main(String[] args) {
        Scanner obj = new Scanner(System.in);
        Cart cart = new Cart();
        System.out.print("Enter number of items: ");
        int n = obj.nextInt();
        String[] items = new String[n];
        double[] prices = new double[n];
        System.out.println("Enter items and prices:");
        for(int i=0;i<n;i++) {
            items[i] = obj.next();
            prices[i] = obj.nextDouble();
        }
    }
}
```

```

    }

    cart.setItems(items);

    cart.setPrices(prices);

    System.out.println("Total cost: " + cart.calculateTotal());

}
}

```

Output:

```

at Main.main(Main.java:24)
C:\Users\kpriy\OneDrive - Amrita Vishwa Vidyapeetham- Chennai Campus\Java\Lab Manual\Encapsulation\Shopping Cart Systemj
avac Main.java
C:\Users\kpriy\OneDrive - Amrita Vishwa Vidyapeetham- Chennai Campus\Java\Lab Manual\Encapsulation\Shopping Cart Systemj
ava Main
Enter number of items: 1
Enter items and prices:
dhal
23
Total cost: 23.0

```

14.d)Temperature Converter

Code:

```

import java.util.Scanner;

class Temperature {

    private double celsius;

    public void setCelsius(double celsius) { this.celsius = celsius; }

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

    public double toKelvin() { return celsius + 273.15; }

}

public class Main {

    public static void main(String[] args) {

        Scanner obj = new Scanner(System.in);

        Temperature temp = new Temperature();

        System.out.print("Enter temperature in Celsius: ");
    }
}

```

```
temp.setCelsius(obj.nextDouble());  
System.out.println("Fahrenheit: " + temp.toFahrenheit());  
System.out.println("Kelvin: " + temp.toKelvin());  
}  
}
```

Output:

```
C:\Users\kpriy\OneDrive - Amrita Vishwa Vidyapeetham- Chennai Campus\Java\Lab Manual\Encapsulation\Temperature Converter  
>javac Main.java  
C:\Users\kpriy\OneDrive - Amrita Vishwa Vidyapeetham- Chennai Campus\Java\Lab Manual\Encapsulation\Temperature Converter  
>java Main  
Enter temperature in Celsius: 23  
Fahrenheit: 73.4  
Kelvin: 296.15
```

15.PACKAGES PROGRAMS

15.a)Library System

Code:

Book.java:

```
package com.library.management;
```

```
public class Book {
```

```
    private String title;
```

```
    private String author;
```

```
    private String isbn;
```

```
    private int publicationYear;
```

```
    public Book(String title, String author, String isbn, int  
    publicationYear) {
```

```
        this.title = title;
```

```
this.author = author;
this.isbn = isbn;
this.publicationYear = publicationYear;
}

// Getters
public String getTitle() { return title; }
public String getAuthor() { return author; }
public String getIsbn() { return isbn; }
public int getPublicationYear() { return publicationYear; }

@Override
public String toString() {
    return String.format("Title: %s\nAuthor: %s\nISBN: %s\nYear:
%d\n",
                        title, author, isbn, publicationYear);
}
}
```

BookManager.java:

```
package com.library.management;

import java.util.ArrayList;
```

```
public class BookManager {  
    private ArrayList<Book> books;  
  
    public BookManager() {  
        books = new ArrayList<>();  
    }  
  
    public void addBook(Book book) {  
        books.add(book);  
        System.out.println("Book added successfully!");  
    }  
  
    public void displayAllBooks() {  
        if (books.isEmpty()) {  
            System.out.println("No books in the library!");  
            return;  
        }  
  
        System.out.println("\nLibrary Books:");  
        for (int i = 0; i < books.size(); i++) {  
            System.out.println("Book #" + (i+1));  
            System.out.println(books.get(i));  
        }  
    }  
}
```



```
}

public void searchBookByTitle(String title) {
    boolean found = false;
    for (Book book : books) {
        if (book.getTitle().equalsIgnoreCase(title)) {
            System.out.println("Book found:");
            System.out.println(book);
            found = true;
        }
    }

    if (!found) {
        System.out.println("No book found with title: " + title);
    }
}

}

}

// File: BookManager.java (in the same package)
package com.library.management;

import java.util.ArrayList;

public class BookManager {
    private ArrayList<Book> books;
```

```
public BookManager() {  
    books = new ArrayList<>();  
}  
  
public void addBook(Book book) {  
    books.add(book);  
    System.out.println("Book added successfully!");  
}  
  
public void displayAllBooks() {  
    if (books.isEmpty()) {  
        System.out.println("No books in the library!");  
        return;  
    }  
  
    System.out.println("\nLibrary Books:");  
    for (int i = 0; i < books.size(); i++) {  
        System.out.println("Book #" + (i+1));  
        System.out.println(books.get(i));  
    }  
}  
  
public void searchBookByTitle(String title) {  
    boolean found = false;
```

```
    for (Book book : books) {  
        if (book.getTitle().equalsIgnoreCase(title)) {  
            System.out.println("Book found:");  
            System.out.println(book);  
            found = true;  
        }  
    }  
  
    if (!found) {  
        System.out.println("No book found with title: " + title);  
    }  
}
```

LibrarySystem.java:

```
package com.library.management;  
  
import java.util.Scanner;  
  
public class LibrarySystem {  
    public static void main(String[] args) {  
        Scanner scanner = new Scanner(System.in);  
        BookManager manager = new BookManager();
```

```
while (true) {  
    System.out.println("\nLibrary Management System");  
    System.out.println("1. Add Book");  
    System.out.println("2. Display All Books");  
    System.out.println("3. Search Book by Title");  
    System.out.println("4. Exit");  
    System.out.print("Enter your choice: ");  
  
    int choice = scanner.nextInt();  
    scanner.nextLine(); // consume newline  
  
    switch (choice) {  
        case 1:  
            System.out.print("Enter book title: ");  
            String title = scanner.nextLine();  
            System.out.print("Enter author: ");  
            String author = scanner.nextLine();  
            System.out.print("Enter ISBN: ");  
            String isbn = scanner.nextLine();  
            System.out.print("Enter publication year: ");  
            int year = scanner.nextInt();  
            manager.addBook(new Book(title, author, isbn, year));  
            break;
```

```
        case 2:
            manager.displayAllBooks();
            break;
        case 3:
            System.out.print("Enter title to search: ");
            String searchTitle = scanner.nextLine();
            manager.searchBookByTitle(searchTitle);
            break;
        case 4:
            System.out.println("Exiting system...");
            scanner.close();
            return;
        default:
            System.out.println("Invalid choice!");
    }
}
}
```

Output:

```
C:\Users\kpriy\OneDrive - Amrita Vishwa Vidyapeetham- Chennai Campus\Java\Lab Manual\Packages Program\User Defined Packages\Library System>java com.library.management.LibrarySystem
```

```
Library Management System
```

1. Add Book
2. Display All Books
3. Search Book by Title
4. Exit

```
Enter your choice: 1
```

```
Enter book title: The Great Gatsby
```

```
Enter author: F. Scott Fitzgerald
```

```
Enter ISBN: 9780743273565
```

```
Enter publication year: 1925
```

```
Book added successfully!
```

```
Library Management System
```

1. Add Book
2. Display All Books
3. Search Book by Title
4. Exit

```
Enter your choice: 1
```

```
Enter book title: To Kill a Mockingbird
```

```
Enter author: Harper Lee
```

```
Enter ISBN: 9780061120084
```

```
Enter publication year: 1960
```

```
Book added successfully!
```

15.b)Bank Account

Code:

AccountManager.java:

```
package bank.simulation.core;
```

```
import java.util.HashMap;
```

```
public class AccountManager {
```

```
    private HashMap<Long, BankAccount> accounts;
```

```
    public AccountManager() {
```

```
        accounts = new HashMap<>();
    }

    public BankAccount createAccount(String name, double
initialBalance) {
        BankAccount account = new BankAccount(name, initialBalance);
        accounts.put(account.getAccountNumber(), account);
        return account;
    }

    public void deposit(long accountNumber, double amount) {
        BankAccount account = accounts.get(accountNumber);
        if (account == null) {
            System.out.println("Account not found!");
            return;
        }
        account.deposit(amount);
        System.out.printf("Deposited %.2f to account %d. New balance:
%.2f\n",
            amount, accountNumber, account.getBalance());
    }

    public void withdraw(long accountNumber, double amount) {
        BankAccount account = accounts.get(accountNumber);
```

```
        if (account == null) {
            System.out.println("Account not found!");
            return;
        }
        try {
            account.withdraw(amount);

            System.out.printf("Withdrew %.2f from account %d. New
balance: %.2f\n",
                amount, accountNumber, account.getBalance());
        } catch (IllegalArgumentException e) {
            System.out.println("Error: " + e.getMessage());
        }
    }

    public void checkBalance(long accountNumber) {
        BankAccount account = accounts.get(accountNumber);
        if (account == null) {
            System.out.println("Account not found!");
            return;
        }
        System.out.println(account);
    }
```



```
public void transferFunds(long sourceAccount, long destAccount,
double amount) {
    BankAccount src = accounts.get(sourceAccount);
    BankAccount dest = accounts.get(destAccount);

    if (src == null || dest == null) {
        System.out.println("One or both accounts not found!");
        return;
    }

    try {
        src.withdraw(amount);
        dest.deposit(amount);

        System.out.printf("Transferred %.2f from account %d to
account %d\n",
            amount, sourceAccount, destAccount);

        System.out.printf("Source account balance: %.2f\n",
src.getBalance());

        System.out.printf("Destination account balance: %.2f\n",
dest.getBalance());
    } catch (IllegalArgumentException e) {
        System.out.println("Transfer failed: " + e.getMessage());
    }
}
```

BankAccount.java:

```
package bank.simulation.core;
```

```
public class BankAccount {
```

```
    private static long accountNumberCounter = 100000000L;
```

```
    private long accountNumber;
```

```
    private String accountHolder;
```

```
    private double balance;
```

```
    public BankAccount(String accountHolder, double balance) {
```

```
        this.accountNumber = ++accountNumberCounter;
```

```
        this.accountHolder = accountHolder;
```

```
        this.balance = balance;
```

```
    }
```

```
    public long getAccountNumber() { return accountNumber; }
```

```
    public String getAccountHolder() { return accountHolder; }
```

```
    public double getBalance() { return balance; }
```

```
    public void deposit(double amount) {
```

```
        if (amount <= 0) {
```

```
        throw new IllegalArgumentException("Deposit amount must  
be positive");
```

```
    }
```

```
    balance += amount;
```

```
}
```

```
public void withdraw(double amount) {
```

```
    if (amount <= 0) {
```

```
        throw new IllegalArgumentException("Withdrawal amount  
must be positive");
```

```
    }
```

```
    if (amount > balance) {
```

```
        throw new IllegalArgumentException("Insufficient funds");
```

```
    }
```

```
    balance -= amount;
```

```
}
```

```
public String toString() {
```

```
    return String.format("Account Number: %d\nHolder:  
%s\nBalance: %.2f",
```

```
        accountNumber, accountHolder, balance);
```

```
}
```

```
}
```

BankApp:

```
package bank.simulation.core;
```

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

```
public class BankApp {
```

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

```
        Scanner scanner = new Scanner(System.in);
```

```
        AccountManager manager = new AccountManager();
```

```
        while (true) {
```

```
            System.out.println("\nBank Account System");
```

```
            System.out.println("1. Create Account");
```

```
            System.out.println("2. Deposit");
```

```
            System.out.println("3. Withdraw");
```

```
            System.out.println("4. Check Balance");
```

```
            System.out.println("5. Transfer Funds");
```

```
            System.out.println("6. Exit");
```

```
            System.out.print("Enter your choice: ");
```

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

```
            switch (choice) {
```

case 1:

```
System.out.print("Enter account holder name: ");
scanner.nextLine(); // consume newline
String name = scanner.nextLine();
System.out.print("Enter initial balance: ");
double balance = scanner.nextDouble();
BankAccount account = manager.createAccount(name,
balance);
System.out.println("Account created successfully!");
System.out.println("Your account number: " +
account.getAccountNumber());
break;
```

case 2:

```
System.out.print("Enter account number: ");
long accNumDeposit = scanner.nextLong();
System.out.print("Enter deposit amount: ");
double depositAmount = scanner.nextDouble();
manager.deposit(accNumDeposit, depositAmount);
break;
```

case 3:

```
System.out.print("Enter account number: ");
long accNumWithdraw = scanner.nextLong();
System.out.print("Enter withdrawal amount: ");
double withdrawAmount = scanner.nextDouble();
```

```
        manager.withdraw(accNumWithdraw, withdrawAmount);
        break;
    case 4:
        System.out.print("Enter account number: ");
        long accNumBalance = scanner.nextLong();
        manager.checkBalance(accNumBalance);
        break;
    case 5:
        System.out.print("Enter source account number: ");
        long sourceAcc = scanner.nextLong();
        System.out.print("Enter destination account number: ");
        long destAcc = scanner.nextLong();
        System.out.print("Enter transfer amount: ");
        double transferAmount = scanner.nextDouble();
        manager.transferFunds(sourceAcc, destAcc,
transferAmount);
        break;
    case 6:
        System.out.println("Exiting bank system...");
        scanner.close();
        return;
    default:
        System.out.println("Invalid choice!");
}
```

```
    }  
  }  
}
```

Output:

```
C:\Users\kpriy\OneDrive - Amrita Vishwa Vidyapeetham- Chennai Campus\Java\Lab Manual\Packag  
ogram\User Defined Packages\Bank Account>java bank.simulation.core.BankApp
```

```
Bank Account System
```

1. Create Account
2. Deposit
3. Withdraw
4. Check Balance
5. Transfer Funds
6. Exit

```
Enter your choice: 1
```

```
Enter account holder name: Priya Kumar
```

```
Enter initial balance: 5000
```

```
Account created successfully!
```

```
Your account number: 10000001
```

```
Bank Account System
```

1. Create Account
2. Deposit
3. Withdraw
4. Check Balance
5. Transfer Funds
6. Exit

```
Enter your choice: 1
```

```
Enter account holder name: Rajesh Patel
```

```
Enter initial balance: 10000
```

```
Account created successfully!
```

```
Your account number: 10000002
```

15.c) Password Checker

Code:

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

```
import java.util.regex.Pattern;
```

```
import java.util.regex.Matcher;
```

```
public class PasswordChecker {  
    public static void main(String[] args) {  
        Scanner scanner = new Scanner(System.in);  
  
        System.out.println("Password Strength Checker");  
        System.out.println("-----");  
  
        System.out.print("Enter your password: ");  
        String password = scanner.nextLine();  
  
        int strength = 0;  
  
        // Check length  
        if (password.length() >= 8) strength++;  
        if (password.length() >= 12) strength++;  
  
        // Check for uppercase letters  
        Pattern upperCase = Pattern.compile("[A-Z]");  
        Matcher matcher = upperCase.matcher(password);  
        if (matcher.find()) strength++;  
  
        // Check for lowercase letters  
        Pattern lowerCase = Pattern.compile("[a-z]");  
        matcher = lowerCase.matcher(password);
```



```
if (matcher.find()) strength++;

// Check for numbers
Pattern digit = Pattern.compile("[0-9]");
matcher = digit.matcher(password);
if (matcher.find()) strength++;

// Check for special characters
Pattern special = Pattern.compile("[^a-zA-Z0-9]");
matcher = special.matcher(password);
if (matcher.find()) strength++;

System.out.println("\nPassword Analysis:");
System.out.println("Length: " + password.length() + "
characters");

String strengthLevel;
if (strength < 3) {
    strengthLevel = "Very Weak";
} else if (strength < 5) {
    strengthLevel = "Weak";
} else if (strength < 7) {
    strengthLevel = "Moderate";
} else if (strength < 9) {
```

```
        strengthLevel = "Strong";
    } else {
        strengthLevel = "Very Strong";
    }

    System.out.println("Strength: " + strengthLevel + " (" + strength
+ "/10)");

    if (strength < 5) {
        System.out.println("\nRecommendations:");
        if (password.length() < 8) {
            System.out.println("- Make your password at least 8
characters long");
        }
        if (!upperCase.matcher(password).find()) {
            System.out.println("- Include at least one uppercase
letter");
        }
        if (!digit.matcher(password).find()) {
            System.out.println("- Include at least one number");
        }
        if (!special.matcher(password).find()) {
            System.out.println("- Include at least one special
character");
        }
    }
}
```

```

    }

    scanner.close();

}
}

```

Output:

```

C:\Users\kpriy\OneDrive - Amrita Vishwa Vidyapeetham- Chennai Campus\Java\Lab Manual\Packages Program\Built In Packagesj
avac PasswordChecker.java

C:\Users\kpriy\OneDrive - Amrita Vishwa Vidyapeetham- Chennai Campus\Java\Lab Manual\Packages Program\Built In Packagesj
ava PasswordChecker
Password Strength Checker
-----
Enter your password: Leena@0012

Password Analysis:
Length: 10 characters
Strength: Moderate (5/10)

```

15.d)Scientific Calculator

Code:

```

import java.util.Scanner;

import java.util.ArrayList;

import java.lang.Math;

public class ScientificCalculator {

    public static void main(String[] args) {

        Scanner scanner = new Scanner(System.in);

        ArrayList<String> history = new ArrayList<>();

        System.out.println("Scientific Calculator");

        System.out.println("-----");
    }
}

```

```
while (true) {  
    System.out.println("\nOperations:");  
    System.out.println("1. Add\t2. Subtract\t3. Multiply\t4.  
Divide");  
    System.out.println("5. Power\t6. Square Root\t7. Sin\t8.  
Cos\t9. Tan");  
    System.out.println("10. View History\t0. Exit");  
    System.out.print("Choose operation: ");  
  
    int choice = scanner.nextInt();  
    if (choice == 0) break;  
  
    double result = 0;  
    String operation = "";  
  
    switch (choice) {  
        case 1:  
            System.out.print("Enter two numbers: ");  
            double a1 = scanner.nextDouble();  
            double b1 = scanner.nextDouble();  
            result = a1 + b1;  
            operation = a1 + " + " + b1 + " = " + result;  
            break;
```

case 2:

```
System.out.print("Enter two numbers: ");  
double a2 = scanner.nextDouble();  
double b2 = scanner.nextDouble();  
result = a2 - b2;  
operation = a2 + " - " + b2 + " = " + result;  
break;
```

case 3:

```
System.out.print("Enter two numbers: ");  
double a3 = scanner.nextDouble();  
double b3 = scanner.nextDouble();  
result = a3 * b3;  
operation = a3 + " * " + b3 + " = " + result;  
break;
```

case 4:

```
System.out.print("Enter two numbers: ");  
double a4 = scanner.nextDouble();  
double b4 = scanner.nextDouble();  
if (b4 == 0) {  
    System.out.println("Error: Division by zero!");  
    continue;  
}
```

```
result = a4 / b4;  
operation = a4 + " / " + b4 + " = " + result;  
break;
```

case 5:

```
System.out.print("Enter base and exponent: ");  
double base = scanner.nextDouble();  
double exp = scanner.nextDouble();  
result = Math.pow(base, exp);  
operation = base + " ^ " + exp + " = " + result;  
break;
```

case 6:

```
System.out.print("Enter number: ");  
double num = scanner.nextDouble();  
if (num < 0) {  
    System.out.println("Error: Square root of negative  
number!");  
    continue;  
}  
result = Math.sqrt(num);  
operation = "√" + num + " = " + result;  
break;
```

case 7:

```
System.out.print("Enter angle in degrees: ");  
double angleSin = scanner.nextDouble();  
result = Math.sin(Math.toRadians(angleSin));  
operation = "sin(" + angleSin + "°) = " + result;  
break;
```

case 8:

```
System.out.print("Enter angle in degrees: ");  
double angleCos = scanner.nextDouble();  
result = Math.cos(Math.toRadians(angleCos));  
operation = "cos(" + angleCos + "°) = " + result;  
break;
```

case 9:

```
System.out.print("Enter angle in degrees: ");  
double angleTan = scanner.nextDouble();  
result = Math.tan(Math.toRadians(angleTan));  
operation = "tan(" + angleTan + "°) = " + result;  
break;
```

case 10:

```
System.out.println("\nCalculation History:");  
for (String item : history) {
```

```

        System.out.println(item);
    }

    continue;

    default:

        System.out.println("Invalid choice!");

        continue;

    }

    System.out.println("Result: " + result);
    history.add(operation);
}

scanner.close();

}

}

```

Output:

```

C:\Users\kpriy\OneDrive - Amrita Vishwa Vidyapeetham- Chennai Campus\Java\Lab Manual\Packages Program\Built In Packagesj
avac ScientificCalculator.java

C:\Users\kpriy\OneDrive - Amrita Vishwa Vidyapeetham- Chennai Campus\Java\Lab Manual\Packages Program\Built In Packagesj
ava ScientificCalculator
Scientific Calculator
-----
Operations:
1. Add 2. Subtract 3. Multiply 4. Divide
5. Power 6. Square Root 7. Sin 8. Cos 9. Tan
10. View History 0. Exit
Choose operation: 1
Enter two numbers: 2
3
Result: 5.0

Operations:
1. Add 2. Subtract 3. Multiply 4. Divide
5. Power 6. Square Root 7. Sin 8. Cos 9. Tan
10. View History 0. Exit
Choose operation: 0

```

16.EXCEPTION HANDLING PROGRAMS

16.a)Age Validator

Code:

```
import java.util.Scanner;

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

public class AgeValidator {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        try {
            System.out.print("Enter your age: ");
            int age = Integer.parseInt(sc.nextLine());
            if(age < 0) throw new InvalidAgeException("Age cannot be negative!");
            if(age > 120) throw new InvalidAgeException("Age seems unrealistic!");
            System.out.println("Valid age: " + age);
        } catch (NumberFormatException e) {
            System.out.println("Error: Please enter a valid number!");
        } catch (InvalidAgeException e) {
```

```
        System.out.println("Error: " + e.getMessage());
    } finally {
        sc.close();
    }
}
}
```

Output:

```
C:\Users\kpriy\OneDrive - Amrita Vishwa Vidyapeetham- Chennai Campus\Java\Lab Manual\Exceptional Handling Program>javac
AgeValidator.java
C:\Users\kpriy\OneDrive - Amrita Vishwa Vidyapeetham- Chennai Campus\Java\Lab Manual\Exceptional Handling Program>java A
geValidator
Enter your age: 23
Valid age: 23
```

16.b) Math Operations

Code:

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

```
public class MathOperations {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        try {
            System.out.print("Enter first number: ");
            double num1 = Double.parseDouble(sc.nextLine());

            System.out.print("Enter second number: ");
            double num2 = Double.parseDouble(sc.nextLine());
```

```
System.out.print("Enter operation (+, -, *, /): ");
char op = sc.nextLine().charAt(0);

switch(op) {
    case '+': System.out.println("Result: " + (num1+num2));
break;

    case '-': System.out.println("Result: " + (num1-num2));
break;

    case '*': System.out.println("Result: " + (num1*num2));
break;

    case '/':
        if(num2 == 0) throw new ArithmeticException("Division
by zero!");

        System.out.println("Result: " + (num1/num2));
        break;

    default: throw new IllegalArgumentException("Invalid
operation!");
}

} catch (NumberFormatException e) {
    System.out.println("Error: Invalid number format!");
} catch (ArithmeticException | IllegalArgumentException e) {
    System.out.println("Error: " + e.getMessage());
} finally {
    sc.close();
}
```

```
}  
  
}
```

Output:

```
C:\Users\kpriy\OneDrive - Amrita Vishwa Vidyapeetham- Chennai Campus\Java\Lab Manual\Exceptional Handling Program>javac  
MathOperations.java  
  
C:\Users\kpriy\OneDrive - Amrita Vishwa Vidyapeetham- Chennai Campus\Java\Lab Manual\Exceptional Handling Program>java M  
athOperations  
Enter first number: 23  
Enter second number: 45  
Enter operation (+, -, *, /): +  
Result: 68.0
```

16.c)File Reader

Code:

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

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

```
import java.io.FileNotFoundException;
```

```
public class FileReader {
```

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

```
        Scanner sc = new Scanner(System.in);
```

```
        try {
```

```
            System.out.print("Enter file path: ");
```

```
            String filePath = sc.nextLine();
```

```
            Scanner fileScanner = new Scanner(new File(filePath));
```

```
            System.out.println("File content:");
```

```
            while(fileScanner.hasNextLine()) {
```

```
                System.out.println(fileScanner.nextLine());
```

```
            }
```

```

        fileScanner.close();
    } catch (FileNotFoundException e) {
        System.out.println("Error: File not found or cannot be read!");
    } catch (SecurityException e) {
        System.out.println("Error: No permission to access the file!");
    } finally {
        sc.close();
    }
}
}

```

Output:

```

C:\Users\kpriy\OneDrive - Amrita Vishwa Vidyapeetham- Chennai Campus\Java\Lab Manual\Exceptional Handling Program>javac FileReader.java
C:\Users\kpriy\OneDrive - Amrita Vishwa Vidyapeetham- Chennai Campus\Java\Lab Manual\Exceptional Handling Program>java FileReader
Enter file path: C:\Users\kpriy\OneDrive - Amrita Vishwa Vidyapeetham- Chennai Campus\Java\Lab Manual\Exceptional Handling Program\FileReader.java
File content:
import java.util.Scanner;
import java.io.File;
import java.io.FileNotFoundException;

public class FileReader {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        try {
            System.out.print("Enter file path: ");
            String filePath = sc.nextLine();

            Scanner fileScanner = new Scanner(new File(filePath));
            System.out.println("File content:");
            while(fileScanner.hasNextLine()) {
                System.out.println(fileScanner.nextLine());
            }
            fileScanner.close();
        } catch (FileNotFoundException e) {
            System.out.println("Error: File not found or cannot be read!");
        } catch (SecurityException e) {
            System.out.println("Error: No permission to access the file!");
        } finally {
            sc.close();
        }
    }
}

```

16.d)Grade Calculator

Code:

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

```

class InvalidGradeException extends Exception {
    public InvalidGradeException(String message) {

```

```
        super(message);  
    }  
}
```

```
public class GradeCalculator {  
    public static void main(String[] args) {  
        Scanner sc = new Scanner(System.in);  
        try {  
            System.out.print("Enter marks obtained (out of 100): ");  
            double marks = Double.parseDouble(sc.nextLine());  
  
            if(marks < 0 || marks > 100)  
                throw new InvalidGradeException("Marks must be between  
0 and 100");  
  
            String grade;  
            if(marks >= 90) grade = "A";  
            else if(marks >= 80) grade = "B";  
            else if(marks >= 70) grade = "C";  
            else if(marks >= 60) grade = "D";  
            else grade = "F";  
  
            System.out.println("Grade: " + grade);  
        } catch (NumberFormatException e) {
```

```

        System.out.println("Error: Please enter a valid number!");
    } catch (InvalidGradeException e) {
        System.out.println("Error: " + e.getMessage());
    } finally {
        sc.close();
    }
}
}
}

```

Output:

```

C:\Users\kpriy\OneDrive - Amrita Vishwa Vidyapeetham- Chennai Campus\Java\Lab Manual\Exceptional Handling Program>javac GradeCalculator.java
C:\Users\kpriy\OneDrive - Amrita Vishwa Vidyapeetham- Chennai Campus\Java\Lab Manual\Exceptional Handling Program>java GradeCalculator
Enter marks obtained (out of 100): 90
Grade: A

```

17. FILE HANDLING PROGRAMS

17.a)File Reader

Code:

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

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

```

public class FileReaderExample {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter the file path to read:");
        String filePath = sc.nextLine();

        try (BufferedReader br = new BufferedReader(new
FileReader(filePath))) {

```

```

        String line;

        System.out.println("\nFile Content:");

        while ((line = br.readLine()) != null) {

            System.out.println(line);

        }

    } catch (IOException e) {

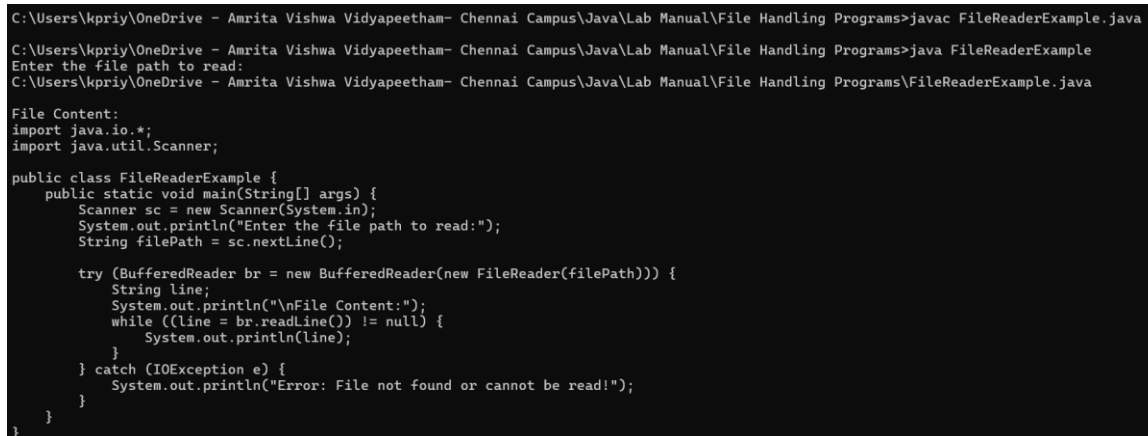
        System.out.println("Error: File not found or cannot be read!");

    }

}
}

```

Output:



```

C:\Users\kpriy\OneDrive - Amrita Vishwa Vidyapeetham- Chennai Campus\Java\Lab Manual\File Handling Programs>javac FileReaderExample.java
C:\Users\kpriy\OneDrive - Amrita Vishwa Vidyapeetham- Chennai Campus\Java\Lab Manual\File Handling Programs>java FileReaderExample
Enter the file path to read:
C:\Users\kpriy\OneDrive - Amrita Vishwa Vidyapeetham- Chennai Campus\Java\Lab Manual\File Handling Programs\FileReaderExample.java

File Content:
import java.io.*;
import java.util.Scanner;

public class FileReaderExample {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter the file path to read:");
        String filePath = sc.nextLine();

        try (BufferedReader br = new BufferedReader(new FileReader(filePath))) {
            String line;
            System.out.println("\nFile Content:");
            while ((line = br.readLine()) != null) {
                System.out.println(line);
            }
        } catch (IOException e) {
            System.out.println("Error: File not found or cannot be read!");
        }
    }
}

```

17.b)File Writer

Code:

```

import java.io.*;

import java.util.Scanner;

public class FileWriterExample {

    public static void main(String[] args) {

```



```
Scanner sc = new Scanner(System.in);

System.out.println("Enter the file path to write:");

String filePath = sc.nextLine();


System.out.println("Enter text to write:");

String text = sc.nextLine();


System.out.println("Append (A) or Overwrite (O)?");

char choice = sc.next().charAt(0);


try (FileWriter fw = new FileWriter(filePath, choice == 'A' ||
choice == 'a')) {

    fw.write(text + "\n");

    System.out.println("Text written successfully!");

} catch (IOException e) {

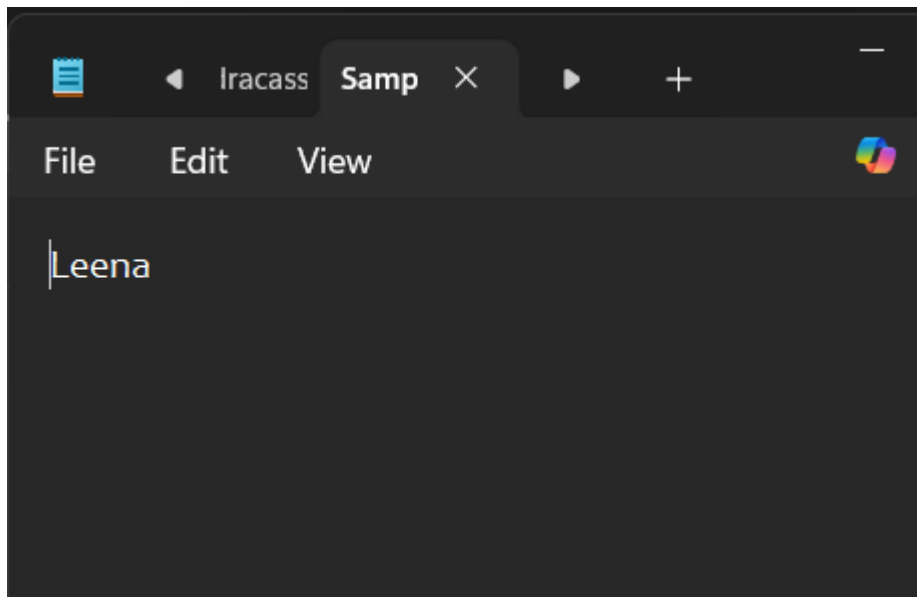
    System.out.println("Error: Unable to write to file!");

}

}
```

Output:

```
C:\Users\kpriy\OneDrive - Amrita Vishwa Vidyapeetham- Chennai Campus\Java\Lab Manual\File Handling Programs>javac FileWriterExample.java
C:\Users\kpriy\OneDrive - Amrita Vishwa Vidyapeetham- Chennai Campus\Java\Lab Manual\File Handling Programs>java FileWriterExample
Enter the file path to write:
C:\Users\kpriy\OneDrive - Amrita Vishwa Vidyapeetham- Chennai Campus\Java\Lab Manual\File Handling Programs\Sample.txt.txt
Enter text to write:
Leena
Append (A) or Overwrite (O)?
o
Text written successfully!
```



17.c)File Deleter

Code:

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

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

```
public class FileDeleter {
```

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

```
        Scanner sc = new Scanner(System.in);
```

```
        System.out.println("Enter the file path to delete:");
```

```
        String filePath = sc.nextLine();
```

```
        File file = new File(filePath);
```

```
        if (file.delete()) {
```

```
            System.out.println("File deleted successfully!");
```

```
        } else {
```

```
        System.out.println("Error: File not found or could not be
deleted!");
```

```
    }
}
}
```

Output:

```
C:\Users\kpriy\OneDrive - Amrita Vishwa Vidyapeetham- Chennai Campus\Java\Lab Manual\File Handling Programs>javac FileDeleter.java
C:\Users\kpriy\OneDrive - Amrita Vishwa Vidyapeetham- Chennai Campus\Java\Lab Manual\File Handling Programs>java FileDeleter
Enter the file path to delete:
C:\Users\kpriy\OneDrive - Amrita Vishwa Vidyapeetham- Chennai Campus\Java\Lab Manual\File Handling Programs\Sample.txt.txt
File deleted successfully!
```

17.d)File Checker

Code:

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

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

```
public class FileChecker {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter the file path to check:");
        String filePath = sc.nextLine();

        File file = new File(filePath);
        if (file.exists()) {
            System.out.println("File exists!");
            System.out.println("Size: " + file.length() + " bytes");
        } else {
```

```
        System.out.println("File does not exist!");  
    }  
}  
}
```

Output:

```
C:\Users\kpriy\OneDrive - Amrita Vishwa Vidyapeetham- Chennai Campus\Java\Lab Manual\File Handling Programs>javac FileChecker.java  
C:\Users\kpriy\OneDrive - Amrita Vishwa Vidyapeetham- Chennai Campus\Java\Lab Manual\File Handling Programs>java FileChecker  
Enter the file path to check:  
C:\Users\kpriy\OneDrive - Amrita Vishwa Vidyapeetham- Chennai Campus\Java\Lab Manual\File Handling Programs\FileChecker.java  
File exists!  
Size: 569 bytes
```