WEEK 9

Question 1: Create a class Vehicle, write a method cost() in this class. Create two classes Bus and Train which have their own display() methods and inherit from Vehicle class. Create objects of Bus and Train class and call cost() and display() methods.

Code: *Vehicle.java*

```
package transport.land.vehicle;
public class Vehicle {
   protected String name;
   protected String startingPoint, destination;
   protected double fairCost;
   public Vehicle(String name, String startingPoint, String destination, double
fairCost) {
       this.name = name;
       this.startingPoint = startingPoint;
       this.destination = destination;
       this.fairCost = fairCost;
   }
   public Vehicle() { this("Unknown", "Unknown", "Unknown", 0); }
   public void cost(){ System.out.println("Fair cost: " + fairCost + " \n"); }
}
```

Bus.java

```
package transport.land.vehicle;
public class Bus extends Vehicle {
   private final int passengerCapacity;
   private final String busType;
   Bus(String name, String start, String dest, double fairCost, int
passengerCapacity, String busType) {
       super(name, start, dest, fairCost);
       this.passengerCapacity = passengerCapacity;
       this.busType = busType;
   }
   public void display(){
       System.out.println("Bus Name: " + name);
       System.out.println("BusType: " + busType);
       System.out.println("Passenger Capacity: " + passengerCapacity);
       System.out.printf("Route: %s to %s. \n", startingPoint, destination);
       super.cost();
   }
}
```

Code:

Train.java

```
package transport.land.vehicle;
public class Train extends Vehicle {
   private int numberOfCoaches;
   private int trainNumber;
   Train(String name, String start, String dest, double fairCost, int
numberOfCoaches, int trainNumber) {
       super(name, start, dest, fairCost);
       this.numberOfCoaches = numberOfCoaches;
       this.trainNumber = trainNumber;
   }
   public void display(){
       System.out.println("Train Name: " + name);
       System.out.println("Train number: " + trainNumber);
       System.out.println("Number of coaches: " + numberOfCoaches);
       System.out.printf("Route: %s to %s. \n", startingPoint, destination);
       super.cost();
   }
}
```

Main.java

```
package transport.land.vehicle;
public class Main {
   public static void main(String[] args) {
       Train Shatabdi = new Train("Shatabdi Express", "Delhi", "Lucknow",
1200.0, 16, 12015);
       Shatabdi.display();
       Bus volvo = new Bus("Volvo Bus", "AMU", "Delhi", 1500.0, 50,  "AC
Sleeper");
       volvo.display();
   }
}
```

Output:

```
"C:\Program Files\Java\OpenJDK\jdk-25\bin\java.exe" "-javaagent:C:\Program
Files\JetBrains\IntelliJ IDEA 2025.2.1\lib\idea_rt.jar=60260" -Dfile.encoding=UTF-8 -
Dsun.stdout.encoding=UTF-8 -Dsun.stderr.encoding=UTF-8 -classpath "D:\Uni
Material\LAB\sem 3\Week 9\Question1\target\classes" transport.land.vehicle.Main
Train Name: Shatabdi Express
Train number: 12015
Number of coaches: 16
Route: Delhi to Lucknow.
Fair cost: 1200.0

Bus Name: Volvo Bus
BusType: AC Sleeper
Passenger Capacity: 50
Route: AMU to Delhi.
Fair cost: 1500.0

Process finished with exit code 0
```

Question 2: Create class University which has data members- name and ranking. Create class Faculty that extends University class has data member- name and method- Details(). Create a new class Department which is derived from Faculty and has data member- name, chairman and method- Details() and Display() where Display() method calls Details() methods of both Faculty and Department class in its body. Create an object of Department class to Display() method and University ranking.

Code: *University.java*

```
package university;
class University {
   protected String uniName;
   protected int ranking;
   public University(String uniName, int ranking) {
      this.uniName = uniName;
      this.ranking = ranking;
   }
   public void showRanking() {
      System.out.println("University Ranking: " + ranking);
   }
}
```

Faculty.java

```
package university;
class Faculty extends University {
   protected String facultyName;
   public Faculty(String uniName, int ranking, String facultyName) {
      super(uniName, ranking);
      this.facultyName = facultyName;
   }
   public void Details() {
      System.out.println("University Name: " + this.uniName);
      System.out.println("Faculty Name: " + facultyName);
   }
}
```

Department.java

```
package university;
class Department extends Faculty {
   private String deptName, chairman;
   public Department(String uniName, int ranking, String facultyName, String
deptName, String chairman) {
      super(uniName, ranking, facultyName);
      this.deptName = deptName;
      this.chairman = chairman;
   }
   public void Details() {
      System.out.println("Department Name: " + deptName);
      System.out.println("Chairman: " + chairman);
   }
   public void Display() {
      super.Details(); this.Details();
   }
}
```

Code: *Main.java*

```
package university;
public class Main {
   public static void main(String[] args) {
      Department CSDept = new Department("Aligarh Muslim University", 9,
            "Science Faculty", "Computer Science Department", "ARF");
      CSDept.Display();
      CSDept.showRanking();
   }
}
```

Output:

```
"C:\Program Files\Java\OpenJDK\jdk-25\bin\java.exe" "-javaagent:C:\Program
Files\JetBrains\IntelliJ IDEA 2025.2.1\lib\idea_rt.jar=57461" -Dfile.encoding=UTF-8 -
Dsun.stdout.encoding=UTF-8 -Dsun.stderr.encoding=UTF-8 -classpath "D:\Uni
Material\LAB\sem 3\Week 9\Question2\target\classes" university.Main

University Name: Aligarh Muslim University

Faculty Name: Science Faculty

Department Name: Computer Science Department

Chairman: ARF

University Ranking: 9


Process finished with exit code 0
```

Question 3: Create class Account (Data members- Id, Account_holder_name, Address; Methodsdeposit(), withdraw()). Create two static methods in Account calculateSimpleInterest() and calculateCompoundInterest() and implement them.

Code: *Main.java*

```
package bank.account;
public class Main {
   public static void main(String[] args) {
      Account acc1 = new Account(151, "Lalu", "Champaran, Bihar", 500);
      acc1.deposit(2000);
      acc1.withdraw(1500);
      double si = Account.calculateSimpleInterest(10000, 5, 2);
      double ci = Account.calculateCompoundInterest(10000, 5, 2);
      System.out.printf("Simple Interest = %.2f", si);
      System.out.printf("Compound Interest = %.2f" , ci);
      acc1.details();
   }
}
```

Code:

Account.java

```
package bank.account;
public class Account {
    private int id;
    private String accountHolderName, address;
    private double balance;
    public Account(int id, String accountHolderName, String address, double balance) {
        this.id = id;
        this.accountHolderName = accountHolderName;
        this.address = address;
        this.balance = balance;
    }
    public void deposit(double amount) {
        if (amount > 0) {
            balance += amount;
            System.out.println(amount + " deposited! New Balance = " + balance);
        } else
            System.out.println("Invalid deposit amount.");
    }
    public void withdraw(double amount) {
        if (amount > 0 && amount <= balance) {
            balance -= amount;
            System.out.println(amount + " withdrawn! New Balance = " + balance);
        } else
            System.out.println("Insufficient balance or invalid amount.");
    }
    public static double calculateSimpleInterest(double principal, double rate, double time) {
        return (principal * rate * time) / 100.0;
    }
    public static double calculateCompoundInterest(double principal, double rate, double time) {
        return principal * (Math.pow((1 + rate / 100.0), time) - 1);
    }
    public void details(){
        System.out.println("Id: " + id);
        System.out.println("Account Holder Name: " + accountHolderName);
        System.out.println("Address: " + address);
        System.out.println("Balance: " + balance);
    }
}
```

Output:

```
"C:\Program Files\Java\OpenJDK\jdk-25\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA 2025.2.1\lib\idea_rt.jar=57472" -Dfile.encoding=UTF-8 -Dsun.stdout.encoding=UTF-8 -Dsun.stderr.encoding=UTF-8 -classpath "D:\Uni Material\LAB\sem 3\Week 9\Question3\target\classes" bank.account.Main
2000.0 deposited! New Balance = 2500.0
1500.0 withdrawn! New Balance = 1000.0
Simple Interest = 1000.00Compound Interest = 1025.00Id: 151
Account Holder Name: Lalu
Address: Champaran, Bihar
Balance: 1000.0

Process finished with exit code 0
```

Question 4: Create class Account (Data members- Id, Account_holder_name, Address; Methodsdeposit(), withdraw()). Declare deposit() and withdraw() as abstract methods. Declare Account class as abstract. (Create constructor in Account as well).

Code: *Main.java*

```
package bank.account;
public class Main {
    public static void main(String[] args) {
        SavingsAccount acc1 = new SavingsAccount(101, "Hamzah",
                "Aligarh", 5000);
        acc1.deposit(2000);
        acc1.withdraw(1000);
        acc1.showDetails();
    }
}
```

Code: *Account.java*

```
package bank.account;
abstract class Account {
   protected int id;
   protected String accountHolderName, address;
   protected double balance;
   public Account(int id, String accountHolderName, String address, double balance) {
      this.id = id;
      this.accountHolderName = accountHolderName;
      this.address = address;
      this.balance = balance;
   }
   public abstract void deposit(double amount);
   public abstract void withdraw(double amount);
   public void showDetails() {
      System.out.println("Account ID: " + id);
      System.out.println("Holder Name: " + accountHolderName);
      System.out.println("Address: " + address);
      System.out.println("Balance: " + balance);
   }
}
```

SavingsAccount.java

```
package bank.account;
class SavingsAccount extends Account {
   public SavingsAccount(int id, String accountHolderName, String address, double
balance) {
      super(id, accountHolderName, address, balance);
   }
   public void deposit(double amount) {
      if (amount > 0) {
         balance += amount;
         System.out.println(amount + " deposited. Balance = " + balance);
      } else
         System.out.println("Invalid deposit amount.");
   }
   public void withdraw(double amount) {
      if (amount > 0 && amount <= balance) {
         balance -= amount;
         System.out.println(amount + " withdrawn. Balance = " + balance);
      } else
         System.out.println("Insufficient balance or invalid amount.");
   }
}
```

Output:

```
"C:\Program Files\Java\OpenJDK\jdk-25\bin\java.exe" "-javaagent:C:\Program
Files\JetBrains\IntelliJ IDEA 2025.2.1\lib\idea_rt.jar=57477" -Dfile.encoding=UTF-8 -
Dsun.stdout.encoding=UTF-8 -Dsun.stderr.encoding=UTF-8 -classpath "D:\Uni
Material\LAB\sem 3\Week 9\Question4\target\classes" bank.account.Main

2000.0 deposited. Balance = 7000.0

1000.0 withdrawn. Balance = 6000.0

Account ID: 101

Holder Name: Hamzah

Address: Aligarh

Balance: 6000.0

Process finished with exit code 0
```

Question 5: Create two children of Account- Saving (Data Members-Min_balance; Methodsdisplay(), deposit(), withdraw()) and Current (Data MembersMax_withdrawl_limit; Methods-display(),deposit(), withdraw()) . Create constructors for both classes. Implementation of deposit() and withdraw() should be specific to Saving and Current class. Create objects of Saving and Current class and display them.

Code: *Main.java*

```
package bank.account;
public class Main {
   public static void main(String[] args) {
      SavingsAccount sAcc = new SavingsAccount(101, "Jhingur", "Aligarh", 5000);
      sAcc.display();
      sAcc.deposit(2000);
      sAcc.withdraw(6000);
      sAcc.withdraw(3000);
      System.out.println();
      CurrentAccount cAcc = new CurrentAccount(102, "Chidiya", "Delhi", 350000);
      cAcc.display();
      cAcc.deposit(200000);
      cAcc.withdraw(515000);
      cAcc.withdraw(115600);
   }
}
```

Code: *Account.java*

```
package bank.account;
abstract class Account {
    protected int id;
    protected String accountHolderName;
    protected String address;
    protected double balance;
    public Account(int id, String accountHolderName, String address, double balance) {
        this.id = id;
        this.accountHolderName = accountHolderName;
        this.address = address;
        this.balance = balance;
    }
    public abstract void deposit(double amount);
    public abstract void withdraw(double amount);
    public void showDetails() {
        System.out.println("Account ID: " + id);
        System.out.println("Holder Name: " + accountHolderName);
        System.out.println("Address: " + address);
        System.out.println("Balance: " + balance);
    }
}
```

SavingsAccount.java

```
package bank.account;
class SavingsAccount extends Account {
    private final double minBalance = 1000;
    public SavingsAccount(int id, String accountHolderName, String address, double
balance) {
        super(id, accountHolderName, address, balance);
    }
    @Override
    public void deposit(double amount) {
        if (amount > 0) {
            balance += amount;
            System.out.println(amount + " deposited in Savings Account. Balance = " +
balance);
        }
    }
    @Override
    public void withdraw(double amount) {
        if (amount > 0 && (balance - amount) >= minBalance) {
            balance -= amount;
            System.out.println(amount + " withdrawn from Savings Account. Balance = " +
balance);
        }
```

Code:

SavingsAccount.java

```
          else {
            System.out.println("Cannot withdraw! Minimum balance requirement not
met.");
        }
    }
    public void display() {
        showDetails();
        System.out.println("Minimum Balance: " + minBalance);
    }
}
```

CurrentAccount.java

```
package bank.account;
class CurrentAccount extends Account {
    private final double maxWithdrawalLimit = 500000;
    public CurrentAccount(int id, String accountHolderName, String address, double
balance) {
        super(id, accountHolderName, address, balance);
    }
    @Override
    public void deposit(double amount) {
        if (amount > 0) {
            balance += amount;
            System.out.println(amount + " deposited in Current Account. Balance = " +
balance);
        }
    }
    @Override
    public void withdraw(double amount) {
        if (amount > 0 && amount <= maxWithdrawalLimit) {
            balance -= amount;
            System.out.println(amount + " withdrawn from Current Account. Balance = " +
balance);
        } else {
            System.out.println("Cannot withdraw! Insufficient Balance or Exceeds max
withdrawal limit.");
        }
    }
    public void display() {
        showDetails();
        System.out.println("Maximum Withdrawal Limit: " + maxWithdrawalLimit);
    }
}
```

Output:

```
"C:\Program Files\Java\OpenJDK\jdk-25\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA 2025.2.1\lib\idea_rt.jar=59004" -Dfile.encoding=UTF-8 -Dsun.stdout.encoding=UTF-8 -Dsun.stderr.encoding=UTF-8 -classpath "D:\Uni Material\LAB\sem 3\Week 9\Question5\target\classes" bank.account.Main

Account ID: 101

Holder Name: Jhingur

Address: Aligarh

Balance: 5000.0

Minimum Balance: 1000.0

2000.0 deposited in Savings Account. Balance = 7000.0

6000.0 withdrawn from Savings Account. Balance = 1000.0

Cannot withdraw! Minimum balance requirement not met.


Account ID: 102

Holder Name: Chidiya

Address: Delhi

Balance: 350000.0

Maximum Withdrawal Limit: 500000.0

200000.0 deposited in Current Account. Balance = 550000.0

Cannot withdraw! Insufficient Balance or Exceeds max withdrawal limit.

115600.0 withdrawn from Current Account. Balance = 434400.0


Process finished with exit code 0
```

OPTIONAL

Question 6: Create a class Shape with a method area(). Create two derived classes Rectangle and Circle that extend Shape. Each class should override the area() method to calculate the area of the respective shape. Create objects of Rectangle and Circle and call their area() methods.

Code:

Main.java

```
package shape;
public class Main {
   public static void main(String[] args) {
      Shape rect = new Rectangle(5, 10);
      Shape circle = new Circle(7);
      rect.area();
      circle.area();
   }
}
```

Shape.java

```
package shape;
abstract public class Shape {
   abstract void area();
}
```

Rectangle.java

```
package shape;
class Rectangle extends Shape {
   private double length, breadth;
   Rectangle(double length, double breadth) {
      this.length = length;
      this.breadth = breadth;
   }
   @Override
   public void area() {
      double result = length * breadth;
      System.out.printf("Area of Rectangle: %.4f \n", result);
   }
}
```

Shape.java

```
package shape;
class Circle extends Shape {
   private double radius;
   Circle(double radius) {
      this.radius = radius;
   }
   @Override
   void area() {
      double result = Math.PI * radius * radius;
      System.out.printf("Area of Circle: %.4f \n", result);
   }
}
```

Output:

```
"C:\Program Files\Java\OpenJDK\jdk-25\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA 2025.2.1\lib\idea_rt.jar=58298" -Dfile.encoding=UTF-8 -Dsun.stdout.encoding=UTF-8 -Dsun.stderr.encoding=UTF-8 -classpath "D:\Uni Material\LAB\sem 3\Week 9\Question6\target\classes" shape.Main

Area of Rectangle: 50.0000

Area of Circle: 153.9380


Process finished with exit code 0
```

Question 7: Create a class Employee with data members: name, salary, and a method showDetails(). Create a class Manager that extends Employee with an additional data member department. Override the showDetails() method in Manager to display all details, including department. Create an object of Manager and call showDetails().

Code: *Main.java*

```java
package neela.kapda;
public class Main {
    public static void main(String[] args) {
        Manager m1 = new Manager("Faraz Ansari", 85000, "Computer Science");
        m1.showDetails();
    }
}
```

Employee.java

```java
package neela.kapda;


public class Employee {
    protected String name;
    protected double salary;


    Employee(String name, double salary) {
        this.name = name;
        this.salary = salary;
    }
    public void showDetails() {
        System.out.println("Employee Name: " + name);
        System.out.println("Salary: " + salary);
    }
}
```

Code: *Manager.java*

```
package neela.kapda;
public class Manager extends Employee {
    private String department;
    Manager(String name, double salary, String department) {
        super(name, salary);
        this.department = department;
    }
    @Override
    public void showDetails() {
        super.showDetails();
        System.out.println("Department: " + department);
    }
}
```

Output:

```
"C:\Program Files\Java\OpenJDK\jdk-25\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA 2025.2.1\lib\idea_rt.jar=53420" -Dfile.encoding=UTF-8 -Dsun.stdout.encoding=UTF-8 -Dsun.stderr.encoding=UTF-8 -classpath "D:\Uni Material\LAB\sem 3\Week 9\Question7\target\classes" neela.kapda.Main

Employee Name: Faraz Ansari

Salary: 85000.0

Department: Computer Science

Process finished with exit code 0
```

Question 8: Create an abstract class Appliance with data members brand, power and abstract methods turnOn() and turnOff(). Create two derived classes WashingMachine and Refrigerator that provide their own implementations of turnOn() and turnOff(). Create objects of WashingMachine and Refrigerator and call their methods.

Code: *Main.java*

```
package appliance;
public class Main {
    public static void main(String[] args) {
        WashingMachine wm = new WashingMachine("Samsung", 1500);
        Refrigerator rf = new Refrigerator("LG", 800);
        wm.turnOn(); wm.turnOff();
        rf.turnOn(); rf.turnOff();
    }
}
```

Code: *Appliance.java*

```
package appliance;
public abstract class Appliance {
   protected String brand;
   protected int power;
   Appliance(String brand, int power) {
      this.brand = brand;
      this.power = power;
   }
   abstract protected void turnOn();
   abstract protected void turnOff();
}
```

Refrigerator.java

```
package appliance;
public class Refrigerator extends Appliance {
   Refrigerator(String brand, int power) {
      super(brand, power);
   }
   @Override
   public void turnOn() {
      System.out.println(brand + " Refrigerator is cooling. Power: " + power + "W");
   }
   @Override
   public void turnOff() {
      System.out.println(brand + " Refrigerator is turned OFF.");
   }
}
```

WashingMachine.java

```
package appliance;
public class WashingMachine extends Appliance {
   WashingMachine(String brand, int power) {
      super(brand, power);
   }
   @Override
   public void turnOn() {
      System.out.println(brand + " Washing Machine is now ON. Power: " + power +
"W");
   }
   @Override
   public void turnOff() {
      System.out.println(brand + " Washing Machine is now OFF.");
   }
}
```

Output:

```
"C:\Program Files\Java\OpenJDK\jdk-25\bin\java.exe" "-javaagent:C:\Program
Files\JetBrains\IntelliJ IDEA 2025.2.1\lib\idea_rt.jar=49601" -Dfile.encoding=UTF-8 -
Dsun.stdout.encoding=UTF-8 -Dsun.stderr.encoding=UTF-8 -classpath "D:\Uni
Material\LAB\sem 3\Week 9\Question8\target\classes" appliance.Main

Samsung Washing Machine is now ON. Power: 1500W

Samsung Washing Machine is now OFF.

LG Refrigerator is cooling. Power: 800W

LG Refrigerator is turned OFF.

Process finished with exit code 0
```

Question 9: Create a class MathOperations with two static methods: findGCD(int a, int b) to calculate the greatest common divisor and findLCM(int a, int b) to calculate the least common multiple. Call these methods without creating an object of MathOperations.

Code: *Main.java*

```
package maths.operations;
public class Main {
    public static void main(String[] args) {
        int a = 48, b = 18;
        int gcd = MathOperations.findGCD(a, b);
        int lcm = MathOperations.findLCM(a, b);
        System.out.println("GCD of " + a + " and " + b + " = " + gcd);
        System.out.println("LCM of " + a + " and " + b + " = " + lcm);
    }
}
```

MathOperations.java

```
package maths.operations;
class MathOperations {
    public static int findGCD(int a, int b) {
        while (b != 0) {
            int temp = b;
            b = a % b;
            a = temp;
        }
        return a;
    }
    public static int findLCM(int a, int b) {
        return (a * b) / findGCD(a, b);
    }
}
```

Output:

```
"C:\Program Files\Java\OpenJDK\jdk-25\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA 2025.2.1\lib\idea_rt.jar=63870" -Dfile.encoding=UTF-8 -Dsun.stdout.encoding=UTF-8 -Dsun.stderr.encoding=UTF-8 -classpath "D:\Uni Material\LAB\sem 3\Week 9\Question9\target\classes" maths.operations.Main
GCD of 48 and 18 = 6
LCM of 48 and 18 = 144


Process finished with exit code 0
```

Question 10: Create a class Student with data members rollNo, name, marks. Add a static variable schoolName. Create static method changeSchoolName() to update schoolName. Demonstrate how the static variable is shared among all objects.

Code: *Main.java*

```
package students;
public class Main {
   public static void main(String[] args) {

      Student s1 = new Student( "Abdul",509, 91.6);
      Student s2 = new Student( "Ahad",560, 90.9);

      System.out.println("Before changing school name:");
      s1.showDetails();
      s2.showDetails();

      Student.changeSchoolName("Saiyyid Hamid Senior Secondary School");

      System.out.println("After changing school name:");
      s1.showDetails();
      s2.showDetails();
   }
}
```

Code:

Student.java

```
package students;

class Student {
    private String name;
    private int rollNo;
    private double marks;

    static String schoolName = "Aligarh Public School";

    Student(String name, int rollNo, double marks) {
        this.rollNo = rollNo;
        this.name = name;
        this.marks = marks;
    }

    public static void changeSchoolName(String newSchoolName) {
        schoolName = newSchoolName;
    }

    public void showDetails() {
        System.out.println("Roll No: " + rollNo);
        System.out.println("Name: " + name);
        System.out.println("Marks: " + marks);
        System.out.println("School: " + schoolName);
        System.out.println();
    }
}
```

Output:

```
"C:\Program Files\Java\OpenJDK\jdk-25\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA 2025.2.1\lib\idea_rt.jar=64250" -Dfile.encoding=UTF-8 -Dsun.stdout.encoding=UTF-8 -Dsun.stderr.encoding=UTF-8 -classpath "D:\Uni Material\LAB\sem 3\Week 9\Question10\target\classes" students.Main
Before changing school name:
Roll No: 509
Name: Abdul
Marks: 91.6
School: Aligarh Public School

Roll No: 560
Name: Ahad
Marks: 90.9
School: Aligarh Public School

After changing school name:
Roll No: 509
Name: Abdul
Marks: 91.6
School: Saiyyid Hamid Senior Secondary School

Roll No: 560
Name: Ahad
Marks: 90.9
School: Saiyyid Hamid Senior Secondary School



Process finished with exit code 0
```