WEEK 9

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.

Main.java

```java
package week9_1;

public class Main {
    public static void main(String[] args) {
        Bus myBus = new Bus();
        myBus.display();
        myBus.cost();

        System.out.println();

        Train myTrain = new Train();
        myTrain.display();
        myTrain.cost();
    }
}
```

Bus.java

```java
package week9_1;

class Bus extends Vehicle {  2 usages
    public void display() {
        System.out.println("This is a Bus.");
    }
}
```

Train.java

```java
package week9_1;

class Train extends Vehicle {  2 usages
    public void display() {
        System.out.println("This is a Train.");
    }
}
```

Vehicle.java

```java
package week9_1;

class Vehicle {  2 usages  2 inheritors
    public void cost() {  2 usages
        System.out.println("The cost of the vehicle is...");
    }
}
```

```
This is a Bus.
The cost of the vehicle is...

This is a Train.
The cost of the vehicle is...
```

WEEK 9

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.

Main.java | Department.java | Faculty.java | University.java

```
1   package week9_2;
2
3   class University {  1 usage  2 inheritors
4       String name;  2 usages
5       int ranking;  2 usages
6
7       public University(String name, int ranking) {  1 usage
8           this.name = name;
9           this.ranking = ranking;
10      }
11  }
12
```

Main.java | Department.java | Faculty.java | University.java

```
1   package week9_2;
2
3   class Faculty extends University {  1 usage  1 inheritor
4       String facultyName;  2 usages
5
6       public Faculty(String uniName, int ranking, String facultyName) {  1 usage
7           super(uniName, ranking);
8           this.facultyName = facultyName;
9       }
10
11      public void Details() {  2 usages  1 override
12          System.out.println("Faculty: " + this.facultyName);
13      }
14  }
15
```

WEEK 9

Main.java × Department.java × Faculty.java University.java

```java
package week9_2;

class Department extends Faculty {  2 usages
    String deptName;  2 usages
    String chairman;  2 usages

    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() {  2 usages
        System.out.println("Department: " + this.deptName);
        System.out.println("Chairman: " + this.chairman);
    }
    public void Display() {  1 usage
        System.out.println("--- Complete Details ---");
        System.out.println("University: " + this.name);
        super.Details();
        this.Details();
    }
}
```

Main.java × Department.java Faculty.java University.java

```java
package week9_2;

public class Main {
    public static void main(String[] args) {
        Department csDept = new Department( uniName: "AMU", ranking: 8,
                facultyName: "F/O Science", deptName: "CS Dept.", chairman: "Mr. A.R. Faridi");

        csDept.Display();

        System.out.println("------------------------");
        System.out.println("University Ranking: " + csDept.ranking);
    }
}
```

```
--- Complete Details ---
University: AMU
Faculty: F/O Science
Department: CS Dept.
Chairman: Mr. A.R. Faridi
------------------------
University Ranking: 8
```

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.

WEEK 9

Main.java Account.java

```java
package week9_3;

class Account {  4 usages
    int id;  2 usages
    String Account_holder_name;  2 usages
    String Address;  2 usages
    private double balance;  7 usages

    public Account(int id, String name, String address, double initialBalance) {  1 usage
        this.id = id;
        this.Account_holder_name = name;
        this.Address = address;
        this.balance = initialBalance;
    }

    public void deposit(double amount) {  1 usage
        if (amount > 0) {
            this.balance += amount;
            System.out.printf("Successfully deposited %.2f. New balance is %.2f\n", amount, this.balance);
        } else {
            System.out.println("Deposit amount must be positive.");
        }
    }

    public void withdraw(double amount) {
        if (amount <= 0) {
            System.out.println("Withdrawal amount must be positive.");
        } else if (this.balance >= amount) {
            this.balance -= amount;
            System.out.printf("Successfully withdrew %.2f. New balance is %.2f\n", amount, this.balance);
        } else {
            System.out.println("Withdrawal failed. Insufficient funds.");
        }
    }

    public void displayDetails() {  2 usages
        System.out.println("--- Account " + this.id + " ---");
        System.out.println("Holder: " + this.Account_holder_name);
        System.out.println("Address: " + this.Address);
        System.out.printf("Current Balance: %.2f\n", this.balance);
        System.out.println("----------------------");
    }

    public static double calculateSimpleInterest(double p, double r, double t) {  1 usage
        return (p * r * t) / 100.0;
    }

    public static double calculateCompoundInterest(double p, double r, double t ) {  1 usage
        double rateAsDecimal = r / 100.0;
        double amount = p * Math.pow(1 + rateAsDecimal, t);
        return amount - p;
    }
}
```

WEEK 9

Main.java × Account.java

```java
package week9_3;

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

        Account account1 = new Account( id: 101, name: "Amit Sharma", address: "Agra, UP", initialBalance: 25000.0);

        System.out.println("Using non-static methods:");
        account1.displayDetails();
        account1.deposit( amount: 10000.0);
        account1.withdraw( amount: 5000.0);
        account1.displayDetails();

        System.out.println();

        System.out.println("Using static methods:");
        double p = 50000;
        double r = 7.5;
        double t = 5;

        double si = Account.calculateSimpleInterest(p, r, t);
        System.out.printf("Simple Interest for %.2f at %.2f%% for %.1f years is: %.2f\n", p, r, t, si);

        double ci = Account.calculateCompoundInterest(p, r, t);
        System.out.printf("Compound Interest for %.2f at %.2f%% for %.1f years (compounded annually) is: %.2f\
    }
}
```

```
Using non-static methods:
--- Account 101 ---
Holder: Amit Sharma
Address: Agra, UP
Current Balance: 25000.00
--------------------
Successfully deposited 10000.00. New balance is 35000.00
Successfully withdrew 5000.00. New balance is 30000.00
--- Account 101 ---
Holder: Amit Sharma
Address: Agra, UP
Current Balance: 30000.00
--------------------

Using static methods:
Simple Interest for 50000.00 at 7.50% for 5.0 years is: 18750.00
Compound Interest for 50000.00 at 7.50% for 5.0 years (compounded annually) is: 21781.47
```

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).

5. Create two children of Account- Saving (Data Members- Min_balance; Methodsdisplay(), deposit(), withdraw()) and

WEEK 9

Current (Data Members- Max_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

Main.java | Account.java × | Current.java | Saving.java

```java
package week9_4;

abstract class Account {  2 usages  2 inheritors
    int id;  7 usages
    String accountHolderName;  3 usages
    String address;  3 usages

    public Account(int id, String name, String address) {  2 usages
        this.id = id;
        this.accountHolderName = name;
        this.address = address;
    }

    public abstract void deposit(double amount);  2 usages  2 implementations
    public abstract void withdraw(double amount);  2 implementations
}
```

Main.java | Account.java | Current.java × | Saving.java

```java
package week9_4;

class Current extends Account {  2 usages
    // Data member specific to Current account
    double maxWithdrawLimit;  2 usages

    /**
     * Constructor for the Current class.
     */
    public Current(int id, String name, String address, double maxWithdrawal) {  1 usage
        super(id, name, address); // Initialize parent class members
        this.maxWithdrawLimit = maxWithdrawal;
    }

    @Override  2 usages
    public void deposit(double amount) {
        System.out.printf("Depositing %.2f into Current Account #%d\n", amount, this.id);
    }

    @Override
    public void withdraw(double amount) {
        System.out.printf("Withdrawing %.2f from Current Account #%d\n", amount, this.id);
    }

    public void display() {
        System.out.println("--- Current Account Details ---");
        System.out.println("Account ID: " + this.id);
        System.out.println("Holder Name: " + this.accountHolderName);
        System.out.println("Address: " + this.address);
        System.out.println("Max Withdrawal Limit: " + this.maxWithdrawLimit);
        System.out.println("-----------------------------");
    }
}
```

WEEK 9

Main.java × Account.java Current.java Saving.java ×

```java
package week9_4;

class Saving extends Account {  2 usages
    // Data member specific to Saving account
    double minBalance;  2 usages

    public Saving(int id, String name, String address, double minBalance) {  1 usage
        super(id, name, address); // Initialize parent class members
        this.minBalance = minBalance;
    }

    @Override  2 usages
    public void deposit(double amount) {
        System.out.printf("Depositing %.2f into Saving Account #%d\n", amount, this.id);
    }

    @Override
    public void withdraw(double amount) {
        System.out.printf("Withdrawing %.2f from Saving Account #%d\n", amount, this.id);
    }

    public void display() {
        System.out.println("--- Saving Account Details ---");
        System.out.println("Account ID: " + this.id);
        System.out.println("Holder Name: " + this.accountHolderName);
        System.out.println("Address: " + this.address);
        System.out.println("Minimum Balance Rule: " + this.minBalance);
        System.out.println("-----------------------------");
    }
}
```

Main.java × Account.java Current.java Saving.java

```java
package week9_4;

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

        Saving savingAccount = new Saving( id: 101,  name: "Shaheer",  address: "AMU",  minBalance: 1000.0);

        Current currentAccount = new Current( id: 202,  name: "Shafin",  address: "AMU",  maxWithdrawal: 100000.0);

        savingAccount.display();
        savingAccount.deposit( amount: 15000);
        savingAccount.withdraw( amount: 4500);

        System.out.println();

        currentAccount.display();
        currentAccount.deposit( amount: 50000);
        currentAccount.withdraw( amount: 25000);
    }
}
```

WEEK 9

```
--- Saving Account Details ---
Account ID: 101
Holder Name: Shaheer
Address: AMU
Minimum Balance Rule: 1000.0
----------------------------
Depositing 15000.00 into Saving Account #101
Withdrawing 4500.00 from Saving Account #101

--- Current Account Details ---
Account ID: 202
Holder Name: Shafin
Address: AMU
Max Withdrawal Limit: 100000.0
-----------------------------
Depositing 50000.00 into Current Account #202
Withdrawing 25000.00 from Current Account #202
```

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.

Main.java | Shape.java | Circle.java | Rectangle.java

```java
package week9_6;

public class Main {
    public static void main(String[] args) {
        Rectangle rect = new Rectangle( length: 12.5,  width: 4.0);
        Circle circle = new Circle( radius: 7.5);

        rect.area();
        circle.area();
    }
}
```

Main.java | Shape.java | Circle.java | Rectangle.java

```java
package week9_6;

class Shape {  2 usages  2 inheritors
    public void area() {  2 usages  2 overrides
        System.out.println("Area of Shape is...");
    }
}
```

WEEK 9

Main.java × | Shape.java | Circle.java × | Rectangle.java

```java
package week9_6;

class Circle extends Shape {  2 usages
    private double radius;  3 usages

    public Circle(double radius) {  1 usage
        this.radius = radius;
    }

    @Override  2 usages
    public void area() {
        double result = Math.PI * radius * radius;
        System.out.printf("Area of Circle is: %.2f\n", result);
    }
}
```

Main.java | Shape.java | Circle.java | Rectangle.java ×

```java
package week9_6;

class Rectangle extends Shape {  2 usages
    private double length;  2 usages
    private double width;  2 usages

    public Rectangle(double length, double width) {  1 usage
        this.length = length;
        this.width = width;
    }

    @Override  2 usages
    public void area() {
        double result = length * width;
        System.out.println("Area of Rectangle is: " + result);
    }
}
```

```
Area of Rectangle is: 50.0
Area of Circle is: 176.71
```

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().

WEEK 9

in.java × Employee.java Manager.java

```
package week9_7;

public class Main {
    public static void main(String[] args) {
        Manager m1 = new Manager( name: "Mohd Nadeem",
                salary: 120000.0,  department: "CS");
        System.out.println("Manager Info:");
        m1.showDetails();
    }
}
```

.java Employee.java × Manager.java

```
package week9_7;

class Employee {  1 usage  1 inheritor
    String name;  2 usages
    double salary;  2 usages

    public Employee(String name, double salary) {  1 usa
        this.name = name;
        this.salary = salary;
    }

    public void showDetails() {  2 usages  1 override
        System.out.println("Name: " + this.name);
        System.out.println("Salary: " + this.salary);
    }
}
```

Main.java Employee.java Manager.java ×

```
1   package week9_7;
2
3   class Manager extends Employee {  2 usages
4       String department;  2 usages
5
6       public Manager(String name, double salary, String department) {
7           super(name, salary);
8           this.department = department;
9       }
10
11      @Override  2 usages
12      public void showDetails() {
13          super.showDetails();
14          System.out.println("Department: " + this.department);
15      }
16  }
```

```
Manager Info:
Name: Mohd Nadeem
Salary: 120000.0
Department: CS
```

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.

WEEK 9

Main.java | Appliance.java | Refrigerator.java | WashingMachine.java

```java
package week9_8;

public class Main {
    public static void main(String[] args) {
        WashingMachine w = new WashingMachine( brand: "Bosch", power: 2200);
        System.out.println("Washing Machine:");
        w.displayInfo();
        w.turnOn();
        w.turnOff();

        System.out.println();

        Refrigerator fridge = new Refrigerator( brand: "Whirlpool", power: 180);
        System.out.println("Refrigerator: ");
        fridge.displayInfo();
        fridge.turnOn();
        fridge.turnOff();
    }
}
```

Main.java | Appliance.java | Refrigerator.java | WashingMachine.java

```java
package week9_8;

abstract class Appliance {  2 usages  2 inheritors
    String brand;  2 usages
    int power;  2 usages

    public Appliance(String brand, int power) {  2 usages
        this.brand = brand;
        this.power = power;
    }

    public abstract void turnOn();  2 usages  2 implementations
    public abstract void turnOff();  2 usages  2 implementations

    public void displayInfo() {  2 usages
        System.out.println("Brand: " + this.brand);
        System.out.println("Power Consumption: " + this.power + "W");
    }
}
```

Main.java | Appliance.java | Refrigerator.java | Washir

```java
package week9_8;

class Refrigerator extends Appliance {  2 usages
    public Refrigerator(String brand, int power) {  1 usag
        super(brand, power);
    }

    @Override  2 usages
    public void turnOn() {
        System.out.println("The fridge is on.");
    }

    @Override  2 usages
    public void turnOff() {
        System.out.println("The fridge is off.");
    }
}
```

WEEK 9

Main.java | Appliance.java | Refrigerator.java | WashingMachin

```java
package week9_8;

class WashingMachine extends Appliance {  2 usages
    public WashingMachine(String brand, int power) {  1 usage
        super(brand, power);
    }

    @Override  2 usages
    public void turnOn() {
        System.out.println("The washing machine is on.");
    }

    @Override  2 usages
    public void turnOff() {
        System.out.println("The washing machine is off.");
    }
}
```

```
Washing Machine:
Brand: Bosch
Power Consumption: 2200W
The washing machine is on.
The washing machine is off.

Refrigerator:
Brand: Whirlpool
Power Consumption: 180W
The fridge is on.
The fridge is off.
```

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.

WEEK 9

MathOperations.java

```java
package week9_9;

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) {
        if (a == 0 || b == 0) {
            return 0;
        }
        return Math.abs(a * b) / findGCD(a, b);
    }
}
```

Main.java

```java
package week9_9;

public class Main {
    public static void main(String[] args) {
        int number1 = 48;
        int number2 = 18;

        int gcd = MathOperations.findGCD(number1, number2);
        System.out.println("GCD: " + gcd);

        int lcm = MathOperations.findLCM(number1, number2);
        System.out.println("LCM: " + lcm);
    }
}
```

```
GCD: 6
LCM: 144
```

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.

Main.java × Student.java

```java
package week9_10;

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

        Student student1 = new Student( rollNo: 508,  name: "Shaheer",  marks: 95.5);
        Student student2 = new Student( rollNo: 548,  name: "Shafin",  marks: 89.0);

        System.out.println("Initial Details of All Students\n");
        student1.displayDetails();
        student2.displayDetails();

        System.out.println();

        Student.changeSchoolName( newName: "JMI");

        System.out.println("Details After Updating School Name\n");
        student1.displayDetails();
        student2.displayDetails();
    }
}
```

WEEK 9

Main.java Student.java

```java
package week9_10;

class Student {  5 usages
    int rollNo;  2 usages
    String name;  2 usages
    double marks;  2 usages
    static String schoolName = "AMU";  2 usages

    public Student(int rollNo, String name, double marks) {  2 usages
        this.rollNo = rollNo;
        this.name = name;
        this.marks = marks;
    }

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

    public void displayDetails() {  4 usages
        System.out.println("Roll No: " + this.rollNo);
        System.out.println("Name: " + this.name);
        System.out.println("Marks: " + this.marks);
        System.out.println("School Name: " + schoolName);
        System.out.println("-------------------------");
    }
}

```

```
Initial Details of All Students

Roll No: 508
Name: Shaheer
Marks: 95.5
School Name: AMU
-------------------------
Roll No: 548
Name: Shafin
Marks: 89.0
School Name: AMU
-------------------------
```

```
Details After Updating School Name

Roll No: 508
Name: Shaheer
Marks: 95.5
School Name: JMI
-------------------------
Roll No: 548
Name: Shafin
Marks: 89.0
School Name: JMI
-------------------------
```