

Question 1: Create class Person (Data Member- name, phone). Create two members inner classes Address (Data Member- House_No, Street, City, State; Method- displayAddr()) and DateOfBirth (Data Member- Day, Month, Year; Method- displayDOB()). Display() is the method of Person class which will display name, address and date of birth of a Person object.

Code:

Person.java

```
package cheap.tcs;
class Person {
    private String name, phone;
    private Address addr;
    private DateOfBirth dob;
    // argument constructor for Person class
    Person(String name, String phone) {
        this.name = name;
        this.phone = phone;
    }
    // method to create Address class object
    public void setAddress(String houseNo, String street, String city, String state)
    {
        addr = new Address(houseNo, street, city, state);
    }
    public void setDOB(int day, int month, int year) {
        dob = new DateOfBirth(day, month, year);
    }
    public void display() {
        System.out.println("Name: " + name);
        System.out.println("Phone: " + phone);
        if (addr != null) addr.displayAddr();
        if (dob != null) dob.displayDOB();
    }
    class Address {
        private String houseNo, street, city, state;

        Address(String houseNo, String street, String city, String state) {
            this.houseNo = houseNo;
            this.street = street;
            this.city = city;
            this.state = state;
        }
    }
}
```

Person.java

```
void displayAddr(){System.out.printf("Address: %s, %s, %s, %s", houseNo,
street, city, state);}
}
class DateOfBirth {
    private int day, month, year;
    DateOfBirth(int day, int month, int year) {
        this.day = day;
        this.month = month;
        this.year = year;
    }
    void displayDOB() {
        System.out.println("DOB: " + day + "/" + month + "/" + year);
    }
}
}
```

Main.java

```
package cheap.tcs;
public class Main {
    public static void main(String[] args) {
        Person p = new Person("Abdul Ahad Ansari", "9876543210");
        // create inner objects through Person's methods
        p.setAddress("12A", "Zakaria Street", "Aligarh", "Uttar Pradesh");
        p.setDOB(14, 11, 2005);
        p.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=63794" -Dfile.encoding=UTF-8 -Dsun.stdout.encoding=UTF-8 -Dsun.stderr.encoding=UTF-8 -classpath "D:\Uni Material\LAB\sem 3\Week 10\Question1\target\classes" cheap.tcs.Main

Name: Abdul Ahad Ansari

Phone: 9876543210

Address: 12A, Zakaria Street, Aligarh, Uttar PradeshDOB: 14/11/2005

Process finished with exit code 0

Question 2: Create class Edible. Within that define two static classes Fruit and Vegetable. Fruit class will have two methods- fruitDetails() is a static method and fruitPackaging() is a non-static method. Vegetable class also has similar methods - vegetableDetails() and vegetablePackaging(). Call all the four methods from main method.

Code: Edible.java

```
package gordon.foodstore;
class Edible {
    static class Fruit {
        static void fruitDetails() {
            System.out.println("Fruit: Apple, Color: Red, Weight: 200g");
        }
        void fruitPackaging() {
            System.out.println("Packing the fruit in a basket");
        }
    }
    static class Vegetable {
        static void vegetableDetails() {
            System.out.println("Vegetable: Carrot, Color: Orange, Weight: 150g");
        }
        void vegetablePackaging() {
            System.out.println("Packing the vegetable in a crate");
        }
    }
}
```

Main.java

```
package gordon.foodstore;
public class Main {
    public static void main(String[] args) {
        Edible.Fruit.fruitDetails();
        Edible.Vegetable.vegetableDetails();
        Edible.Fruit apple = new Edible.Fruit();
        apple.fruitPackaging();
        Edible.Vegetable carrot = new Edible.Vegetable();
        carrot.vegetablePackaging();
    }
}
```

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=55465" -Dfile.encoding=UTF-8 -Dsun.stdout.encoding=UTF-8 -Dsun.stderr.encoding=UTF-8 -classpath "D:\Uni Material\LAB\sem 3\Week 10\Question2\target\classes" gordon.foodstore.Main

Fruit: Apple, Color: Red, Weight: 200g

Vegetable: Carrot, Color: Orange, Weight: 150g

Packing the fruit in a basket

Packing the vegetable in a crate

Process finished with exit code 0

Question 3: Create three different minMaxAdd() methods to calculate minimum, maximum and addition of integers, real numbers and characters.

Code:

Maths.java

```
package maths;
class Maths {
    public static void minMaxAdd(int a, int b) {
        int min = Math.min(a, b);
        int max = Math.max(a, b);
        int sum = a + b;
        System.out.println("Integers: min=" + min + ", max=" + max + ", sum=" +
sum);
    }
    public static void minMaxAdd(double a, double b) {
        double min = Math.min(a, b);
        double max = Math.max(a, b);
        double sum = a + b;
        System.out.println("Real numbers: min=" + min + ", max=" + max + ", sum=" +
sum);
    }
    public static void minMaxAdd(char a, char b) {
        char min = (a < b) ? a : b;
        char max = (a > b) ? a : b;
        int sum = a + b;
        System.out.println("Characters: min=" + min + ", max=" + max + ", sum of
ASCII=" + sum);
    }
}
```

Code:Main.java

```
package maths;
public class Main {
    public static void main(String[] args) {
        Maths.minMaxAdd(10, 20);
        Maths.minMaxAdd(3.5, 7.2);
        Maths.minMaxAdd('A', 'M');
    }
}
```

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=60225" -Dfile.encoding=UTF-8 -Dsun.stdout.encoding=UTF-8 -Dsun.stderr.encoding=UTF-8 -classpath "D:\Uni Material\LAB\sem 3\Week 10\Question3\target\classes" maths.Main

Integers: min=10, max=20, sum=30

Real numbers: min=3.5, max=7.2, sum=10.7

Characters: min=A, max=M, sum of ASCII=142

Process finished with exit code 0

Question 4: Create a class `ObjectOriented` which has methods- `abstraction()`, `polymorphism()` and `inheritance()`. Create a class `JavaLanguage` which inherits from `ObjectOriented` class and has its own methods `persistence()` and `interfaces()`. Create an object of `JavaLanguage` class to access all of its own and parent's methods

Code:ObjectOriented.java

```
package lang.oops;
public class ObjectOriented {
    void abstraction() {
        System.out.println("Abstraction: Hiding implementation details and showing functionality.");
    }
    void polymorphism() {
        System.out.println("Polymorphism: Ability of an object to take many forms.");
    }
    void inheritance() {
        System.out.println("Inheritance: Mechanism to acquire properties of another class.");
    }
}
```

Code:JavaLanguage.java

```
package lang.oops;
public class JavaLanguage extends ObjectOriented {
    void persistence() {
        System.out.println("Persistence: Java supports saving object state for long-term
use.");
    }
    void interfaces() {
        System.out.println("Interfaces: Java uses interfaces to achieve multiple
inheritance.");
    }
}
```

Main.java

```
package lang.oops;
public class Main {
    public static void main(String[] args) {
        JavaLanguage javaObj = new JavaLanguage();
        javaObj.abstraction();
        javaObj.polymorphism();
        javaObj.inheritance();
        javaObj.persistence();
        javaObj.interfaces();
    }
}
```

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=51080" -Dfile.encoding=UTF-8 -Dsun.stdout.encoding=UTF-8 -Dsun.stderr.encoding=UTF-8 -classpath "D:\Uni Material\LAB\sem 3\Week 10\Question4\target\classes" lang.oops.Main

Abstraction: Hiding implementation details and showing functionality.

Polymorphism: Ability of an object to take many forms.

Inheritance: Mechanism to acquire properties of another class.

Persistence: Java supports saving object state for long-term use.

Interfaces: Java uses interfaces to achieve multiple inheritance.

Process finished with exit code 0

Question 5: In previous question, create a new class C++ which also inherits from ObjectOriented class and has its own methods-template() and friendFunction(). Create an object of C++ class to access all of its own and parent's methods.

Code: ObjectOriented.java

```
package lang.oops;
public class ObjectOriented {
    void abstraction() {
        System.out.println("Abstraction: Hiding implementation details and showing
functionality.");
    }
    void polymorphism() {
        System.out.println("Polymorphism: Ability of an object to take many forms.");
    }
    void inheritance() {
        System.out.println("Inheritance: Mechanism to acquire properties of another
class.");
    }
}
```

CPlusPlus.java

```
package lang.oops;
public class CPlusPlus extends ObjectOriented {
    void templateMethod() {
        System.out.println("Template Method: Allows writing generic and reusable
code in C++.");
    }
    void friendFunction() {
        System.out.println("Friend Function: Allows non-member function to access
private members.");
    }
}
```

Main.java

```
package lang.oops;
public class Main {
    public static void main(String[] args) {
        CPlusPlus cpp = new CPlusPlus();
        cpp.abstraction();
        cpp.polymorphism();
        cpp.inheritance();
        cpp.templateMethod();
        cpp.friendFunction();
    }
}
```

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=49883" -Dfile.encoding=UTF-8 -Dsun.stdout.encoding=UTF-8 -Dsun.stderr.encoding=UTF-8 -classpath "D:\UniMaterial\LAB\sem 3\Week 10\Question5\target\classes" lang.oops.Main

Abstraction: Hiding implementation details and showing functionality.

Polymorphism: Ability of an object to take many forms.

Inheritance: Mechanism to acquire properties of another class.

Template Method: Allows writing generic and reusable code in C++.

Friend Function: Allows non-member function to access private members.

Process finished with exit code 0

Question 6: 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: *Main.java*

```
package central.uni;
public class Main {
    public static void main(String[] args) {
        Department dept = new Department("Aligarh Muslim University", 8,
            "Science Faculty", "Computer Science", "Arman Rasool Faridi");
        dept.Display();
        System.out.println("University Ranking: " + dept.ranking);
    }
}
```


Code: University.java

```
package central.uni;
public class University {
    protected String uniName;
    protected int ranking;
    University(String uniName, int ranking) {
        this.uniName = uniName;
        this.ranking = ranking;
    }
}
```

Faculty.java

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

Department.java

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

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=50070" -Dfile.encoding=UTF-8 -Dsun.stdout.encoding=UTF-8 -Dsun.stderr.encoding=UTF-8 -classpath "D:\Uni Material\LAB\sem 3\Week 10\Question6\target\classes" central.uni.Main

University: Aligarh Muslim University

Faculty Name: Science Faculty

Department Name: Computer Science

Chairman: Arman Rasool Faridi

University Ranking: 8

Process finished with exit code 0

OPTIONAL

Question 7: Create a class Employee (Data Members – empName, empId). Create two-member inner class:

- Salary (Data Members - basic, hra, pf; Method - displaySalary() to print salary details).
- JoiningDate (Data Members - day, month, year; Method - displayJoiningDate() to print joining date).

In the Employee class, create a method displayEmployee that prints the employee's name, ID, salary details and joining date.

Code: Main.java

```
package neela.kapda;
public class Main {
    public static void main(String[] args) {
        Employee emp = new Employee("Abdul Ansari", 547);
        emp.setSalary(420000, 30000, 25000);
        emp.setJoiningDate(15, 8, 2022);
        emp.displayEmployee();
    }
}
```

Code:Employee.java

```

package neela.kapda;
public class Employee {
    private String empName;
    private int empID;
    private Salary salary;
    private JoiningDate joiningDate;
    Employee(String empName, int empID) {
        this.empName = empName; this.empID = empID;
    }
    public void setSalary(double basic, double hra, double pf) {
        this.salary = new Salary(basic, hra, pf);
    }
    public void setJoiningDate(int day, int month, int year) {
        this.joiningDate = new JoiningDate(day, month, year);
    }
    class Salary {
        private double basic, hra, pf;
        Salary(double basic, double hra, double pf) {
            this.basic = basic; this.hra = hra; this.pf = pf;
        }
        void displaySalary() {
            System.out.println("Salary Details: ");
            System.out.println("Basic: " + basic);
            System.out.println("HRA: " + hra);
            System.out.println("PF: " + pf);
        }
    }
    class JoiningDate {
        int day, month, year;
        JoiningDate(int day, int month, int year) {
            this.day = day; this.month = month; this.year = year;
        }
        void displayJoiningDate() {
            System.out.println("Joining Date: " + day + "/" + month + "/" + year);
        }
    }
    void displayEmployee() {
        System.out.println("Employee Name: " + empName);
        System.out.println("Employee ID: " + empID);
        if (salary != null) salary.displaySalary();
        if (joiningDate != null) joiningDate.displayJoiningDate();
    }
}

```

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=55687" -Dfile.encoding=UTF-8 -Dsun.stdout.encoding=UTF-8 -Dsun.stderr.encoding=UTF-8 -classpath "D:\Uni Material\LAB\sem 3\Week 10\Question7\target\classes" neela.kapda.Main

Employee Name: Abdul Ansari

Employee ID: 547

Salary Details:

Basic: 420000.0

HRA: 30000.0

PF: 25000.0

Joining Date: 15/8/2022

Process finished with exit code 0

Question 8: Create a class Shape with overloaded methods area():

- area(int side) – calculates area of a square.
- area(int length, int breadth) – calculates area of a rectangle.
- area(double radius) – calculates area of a circle.

Code:

Main.java

```
package area.calculator;
public class Main {
    public static void main(String[] args) {
        AreaCalculator s = new AreaCalculator();

        System.out.println("Area of Square (side=5): " + s.area(5));
        System.out.println("Area of Rectangle (length=4, breadth=6): " + s.area(4, 6));
        System.out.println("Area of Circle (radius=3.5): " + s.area(3.5));
    }
}
```

AreaCalculator.java

```
package area.calculator;
public class AreaCalculator {
    double area(int side) { return side * side; }
    double area(int length, int breadth) { return length * breadth; }
    double area(double radius) { return Math.PI * radius * radius; }
}
```

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=58515" -Dfile.encoding=UTF-8 -Dsun.stdout.encoding=UTF-8 -Dsun.stderr.encoding=UTF-8 -classpath "D:\UniMaterial\LAB\sem 3\Week 10\Question8\target\classes" area.calculator.Main

Area of Square (side=5): 25.0

Area of Rectangle (length=4, breadth=6): 24.0

Area of Circle (radius=3.5): 38.48451000647496

Process finished with exit code 0

Question 9: Create a class Vehicle with a method run(). Create subclasses Bike and Car that override the run() method. In the main() method, use a reference of Vehicle to call run() for objects of Bike and Car.

Code: Main.java

```
package gaddi;
public class Main {
    public static void main(String[] args) {
        Vehicle v;
        v = new Bike();
        v.run();
        v = new Car();
        v.run();
    }
}
```

Vehicle.java

```
package gaddi;
public class Vehicle {
    void run() { System.out.println("Vehicle is running..."); }
}
class Bike extends Vehicle {
    @Override
    void run() { System.out.println("Bike is running..."); }
}
class Car extends Vehicle {
    @Override
    void run() { System.out.println("Car is running..."); }
}
```

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=52144" -Dfile.encoding=UTF-8 -Dsun.stdout.encoding=UTF-8 -Dsun.stderr.encoding=UTF-8 -classpath "D:\UniMaterial\LAB\sem 3\Week 10\Question9\target\classes" gaddi.Main

Bike is running...

Car is running...

Process finished with exit code 0