

LAB REPORT



ULAB
UNIVERSITY OF LIBERAL ARTS
BANGLADESH

OBJECT ORIENTED PROGRAMMING

LAB 05

SUBMITTED BY:

Piyas sarkar

223014024

Section 04

Question number 1:

Create a class called Person with properties such as name, age, gender, and address. Use constructor overloading, method overloading, and the 'this keyword'.

Question number 2:

Create an employee class with properties such as name, ID, salary, and designation. Use constructor overloading, method overloading, and the 'this keyword'.

Solution of Q1 AND Q2:

```
1  package oopd5;
2
3  public class App
4  {
5      public static void main( String[] args )
6      {
7          System.out.println("showing person");
8          Person person1 = new Person("mojid", 25,"male", "battery goli");
9          Person person2 = new Person("kislu", 30, "male", "puran bari");
10         System.out.println("Before overloading");
11         person1.display();
12         person2.display();
13
14         System.out.println();
15
16         person1.changePerson("mojidul");
17         person2.changePerson(35);
18         System.out.println("After overloading");
19         person1.display();
20         person2.display();
21
22         System.out.println("showing employee");
23         Employee employee1 = new Employee("John Doe", 35, "male", "123 Main St", 1001, 5000.0, "Manager");
24         Employee employee2 = new Employee("Jane Smith", 28, "female", "456 Elm St", 1002, 4000.0, "Supervisor");
25
```



```
1  System.out.println("Before overloading");
2      employee1.display();
3      employee2.display();
4
5  System.out.println();
6
7      employee1.changeEmployee(1003);
8      employee2.changeEmployee(4500.0);
9  System.out.println("After overloading");
10     employee1.display();
11     employee2.display();
12
13
14     }
15 }
16
17 class Person{
18     private String name;
19     private int age;
20     private String gender;
21     private String address;
22
```



```
1  // Constructor
2  public Person(String name, int age, String gender, String address){
3      this.name = name;
4      this.age = age;
5      this.gender = gender;
6      this.address = address;
7  }
8  // constructor overloading
9  public Person(String name, int age, String gender){
10     this(name, age, gender, "");
11 }
12 public void display(){
13     System.out.println("Name: " + name);
14     System.out.println("Age: " + age);
15     System.out.println("Gender: " + gender);
16     System.out.println("Address: " + address);
17 }
18 // method overloading of name property
19 public void changePerson(String name){
20     this.name = name;
21 }
22 public void changePerson(int age){
23     this.age = age;
24 }
25 }
26
27 class Employee {
28     private Person person;
29     private int id;
30     private double salary;
31     private String designation;
```

```
1
2 // Constructor
3 public Employee(String name, int age, String gender, String address, int id, double salary, String designation) {
4     this.person = new Person(name, age, gender, address);
5     this.id = id;
6     this.salary = salary;
7     this.designation = designation;
8 }
9
10 public void display() {
11     person.display();
12     System.out.println("ID: " + id);
13     System.out.println("Salary: " + salary);
14     System.out.println("Designation: " + designation);
15 }
16
17 public void changeEmployee(int id) {
18     this.id = id;
19 }
20
21 public void changeEmployee(double salary) {
22     this.salary = salary;
23 }
24
25 public void changeEmployee(String designation) {
26     this.designation = designation;
27 }
28 }
```

Explanation:

In this code, constructor overloading is demonstrated in the `Person` class with two constructors: one that accepts all properties (name, age, gender, and address) and another that omits the address. Method overloading is shown in both classes, where `changePerson` and `changeEmployee` methods modify different properties based on the parameter type (e.g., name, age, or ID). This highlights how methods with the same name can perform other actions depending on the parameters provided.

Code Outputs:

showing employee

Before overloading

Name: John Doe

Age: 35

Genedr: male

Address: 123 Main St

ID: 1001

Salary: 5000.0

Designation: Manager

Name: Jane Smith

Age: 28

Genedr: female

After overloading

Name: John Doe

Age: 35

Genedr: male

Address: 123 Main St

ID: 1003

Salary: 5000.0

Designation: Manager

Name: Jane Smith

Age: 28

Genedr: female

Address: 456 Elm St

ID: 1002

Salary: 4500.0

Designation: Supervisor

showing person

Before overloading

Name: mojid

Age: 25

Genedr: male

Address: battery goli

Name: kisl

Age: 30

Genedr: male

Address: puran bari

After overloading

Name: mojidul

Age: 25

Genedr: male

Address: battery goli

Name: kisl

Age: 35

Genedr: male

Address: puran bari