

Course Outcome 3(CO3)

CO3: Implement object-oriented concepts like inheritance, overloading and interfaces

Syllabus: Inheritance, Interfaces, Abstract classes, Dynamic Method Dispatch, String

1. Area of different shapes using overloaded functions
2. Create a class 'Employee' with data members Empid, Name, Salary, Address and constructors to initialize the data members. Create another class 'Teacher' that inherit the properties of class employee and contain its own data members department, Subjects taught and constructors to initialize these data members and also include display function to display all the data members. Use array of objects to display details of N teachers.
3. Create a class 'Person' with data members Name, Gender, Address, Age and a constructor to initialize the data members and another class 'Employee' that inherits the properties of class Person and also contains its own data members like Empid, Company_name, Qualification, Salary and its own constructor. Create another class 'Teacher' that inherits the properties of class Employee and contains its own data members like Subject, Department, Teacherid and also contain constructors and methods to display the data members. Use array of objects to display details of N teachers.
4. Write a program has class Publisher, Book, Literature and Fiction. Read the information and print the details of books from either the category, using inheritance.
5. Create classes **Student** and **Sports**. Create another class **Result inherited from Student and Sports**. Display the academic and sports score of a student.
6. Create an **interface** having prototypes of functions **area ()** and **perimeter()**. Create **two classes Circle and Rectangle** which **implements the above interface**. Create a **menu-driven program** to find the area and perimeter of objects.
7. Prepare bill with the given format using the calculate method from the interface.

Order No.

Date :

Product Id	Name	Quantity	unit price	Total
101	A	2	25	50
102	B	1	100	100
			Net. Amount	150

1. CO3.1 Area of different shapes using overloaded functions

/*Method Overloading allows different methods to have the same name, but different signatures where the signature can differ by the number of input parameters or type of input parameters, or a mixture of both */

//Overload.java

import java.util.Scanner;

public class Overload

{

void area (double r)

{

double Ar = 3.14*r*r;

System.out.println ("Area of Circle = "+Ar);

}

void area (int len, int br)

{

int Ar = len*br;

System.out.println("Area of Rectangle = "+ Ar);

}

void area (int l, int b, int h)

{

int Ar = l*b*h;

System.out.println("Area of Cuboid = "+ Ar);

}

public static void main(String[] args)

{

Scanner sc = new Scanner(System.in);

Overload obj1 = new Overload() ;

System.out.println ("Enter the Radius");

int r = sc.nextInt();

obj1.area(r);

System.out.println("Enter the Length & Breath ");

int len = sc.nextInt();

int br = sc.nextInt();

obj1.area(len,br);

System.out.println("Enter the Length, Breath & Height ");

int l = sc.nextInt();

int b = sc.nextInt();

int h = sc.nextInt();

obj1.area (l, b, h);

}

}

/* E:\S2 JAVA LAB 2024\JavaRec>javac Overload.java

E:\S2 JAVA LAB 2024\JavaRec>java Overload

Enter the Radius

2

Area of Circle = 12.56

Enter the Length

3

Enter the Breath

4

Area of Rectangle = 12

Enter the Length

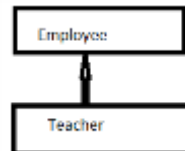
3

Enter the Breath

4

Enter the Height */

2. CO3.2 Create a class **Employee** with **data members Empid, Name, Salary, Address** and **constructors to initialize the data members**. Create **another class 'Teacher'** that inherit the properties of class employee and contain its own **data members department, Subjects taught** and **constructors to initialize these data members** and also include **display function to display all the data members**. Use **array of objects** to display details of N teachers. //TeacherArrOb.java



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

```
/* Create a class Employee with data members: Empid , Name, Salary, Address & constructors to initialize the data members*/
```

```
class Employee
```

```
{
    protected int empId;
    protected String name;
    protected double salary;
    protected String address;
    public Employee(int empId, String name, double salary, String address)
    {
        this.empId = empId;
        this.name = name;
        this.salary = salary;
        this.address = address;
    }
}
```

```
/*Create another class Teacher that inherits the properties of class Employee & contains its own data members department, Subjects taught and constructors to initialize these data members and include a display function to display all the data members */
```

```
class Teacher extends Employee
```

```
{
    private String dept;
    private String sub;
    public Teacher(int empId, String name, double salary, String address, String dept, String sub)
    { super (empId, name, salary, address);
        this.dept = dept;
        this.sub = sub;
    }
    public void display()
    {
        System.out.println("Employee ID: " + empId);
        System.out.println("Name: " + name);
        System.out.println("Salary: " + salary);
        System.out.println("Address: " + address);
    }
}
```

```

        System.out.println("Department: " + dept);
        System.out.println("Subjects Taught: " + sub);
        System.out.println();
    }
}
//Use array of objects to display details of N teachers
public class TeacherArrOb
{
    public static void main(String[] args)
    {
        Scanner sc = new Scanner (System.in);
        System.out.print ("Enter the number of teachers: ");
        int N = sc.nextInt();
        sc.nextLine(); // Consume newline
        Teacher[ ] teachers = new Teacher[N];
        System.out.println("Enter details for Teacher ");
        for (int i = 0; i < N; i++)
        {
            System.out.print("Employee ID: ");
            int empId = sc.nextInt();
            sc.nextLine(); // Consume newline
            System.out.print("Name: ");
            String name = sc.nextLine();
            System.out.print("Salary: ");
            double salary = sc.nextDouble();
            sc.nextLine(); // Consume newline
            System.out.print("Address: ");
            String address = sc.nextLine();
            System.out.print("Department: ");
            String dept = sc.nextLine();
            System.out.print("Subjects Taught: ");
            String sub = sc.nextLine();
            teachers[i] = new Teacher(empId, name, salary, address, dept, sub);
        }
        System.out.println("\nDetails of all teachers:");
        for (Teacher teacher: teachers)
            teacher.display();
    }
}

```

```

/*    for .Each
    for (type var : array)
    {
        statements using var;
    }
*/

```

```

/*E:\S2 JAVA LAB 2024\JavaRec>javac TeacherArrOb.java
E:\S2 JAVA LAB 2024\JavaRec>java TeacherArrOb

```

Enter the number of teachers: 2

Enter details for Teacher

Employee ID: 100

Name: Fariza

Salary: 30000

Address: asaddsfdsd

Department: MCA

Subjects Taught: OB

Employee ID: 101

Name: Anu

Salary: 25000

Address: sdfdsf

Department: MSc

Subjects Taught: DAA

Details of all teachers:

Employee ID: 100

Name: Fariza

Salary: 30000.0

Address: asaddsfdsd

Department: MCA

Subjects Taught: OB

Employee ID: 101

Name: Anu

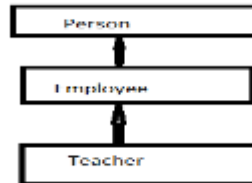
Salary: 25000.0

Address: sdfdsf

Department: MSc

Subjects Taught: DAA */

3. CO3.3 Create a **class 'Person'** with **data members Name, Gender, Address, Age** and a **constructor to initialize the data members** and **another class 'Employee'** that inherits the properties of class Person and also contains its own data members like **Empid, Company_name, Qualification, Salary** and **its own constructor**. Create **another class 'Teacher'** that inherits the properties of class Employee and contains its own data members like **Subject, Department, Teacherid** and also contain **constructors** and **methods** to display the data members. Use **array of objects** to display details of N teachers. //MultilevelTeacher.java



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

```
class Person
```

```
{
    {
        String name, gender, addr;
        int age;
        Person (String name, String gender, String addr, int age)
        {
            this.name = name;
            this.gender = gender;
            this.addr = addr;
            this.age = age;
        }
    }
}
```

```
class Employee extends Person
```

```
{
    int empid;
    String compName, qual;
    double salary;
    Employee (String name, String gender, String addr, int age,  

int empid, String compName, String qual, double salary)
```

```
{
    super(name, gender, addr, age);
    this.empid = empid;
    this.compName = compName;
    this.qual = qual;
    this.salary = salary;
}
```

```
class Teacher extends Employee
```

```
{
```

```

String sub, dept;
int Tid;
Teacher(String name, String gender, String addr, int age, int empid,
String compName, String qual, double salary, String sub,
String dept, int Tid)
{
    super(name, gender, addr, age, empid, compName, qual, salary);
    this.sub = sub;
    this.dept = dept;
    this.Tid = Tid;
}
void display()
{
    System.out.println ("Name: " + name);
    System.out.println ("Gender: " + gender);
    System.out.println ("Address: " + addr);
    System.out.println("Age: " + age);
    System.out.println("Employee ID: " + empid);
    System.out.println("Company Name: " + compName);
    System.out.println("Qualification: " + qual);
    System.out.println("Salary: " + salary);
    System.out.println("Subject: " + sub);
    System.out.println("Department: " + dept);
    System.out.println("Teacher ID: " + Tid);
}
}
public class MultilevelTeacher
{
    public static void main(String[] args)
    {
        Scanner sc = new Scanner (System.in);
        System.out.print("Enter the number of teachers: ");
        int n = sc.nextInt();
        Teacher[] teachers = new Teacher[n];
        System.out.println("Enter details for teacher ");
        for (int i = 0; i < n; i++)
        {
            System.out.print("Name: ");
            String name = sc.next();
            System.out.print("Gender: ");
            String gender = sc.next();
            System.out.print("Address: ");
            String addr = sc.next();
            System.out.print("Age: ");
            int age = sc.nextInt();
            System.out.print("Employee ID: ");

```



```

        int empid = sc.nextInt();
        System.out.print("Company Name: ");
        String compName = sc.next();
        System.out.print("Qualification: ");
        String qual = sc.next();
        System.out.print("Salary: ");
        double salary = sc.nextDouble();
        System.out.print("Subject: ");
        String sub = sc.next();
        System.out.print("Department: ");
        String dept = sc.next();
        System.out.print("Teacher ID: ");
        int Tid = sc.nextInt();
        teachers[i] = new Teacher(name, gender, addr, age, empid,
compName, qual, salary, sub, dept, Tid);
    }
    System.out.println("\nDetails of Teachers:");
    for (Teacher i: teachers)
    {
        i.display ( );
        System.out.println();
    }
}
}
/*
E:\S2 JAVA LAB 2024\JavaRec>javac MultilevelTeacher.java
E:\S2 JAVA LAB 2024\JavaRec>java MultilevelTeacher
Enter the number of teachers: 2
Enter details for teacher
Name: ammu
Gender: f
Address: qqweq
Age: 34
Employee ID: 10
Company Name: icet
Qualification: mca
Salary: 50000
Subject: cn
Department: mca
Teacher ID: 110
Name: feby
Gender: m
Address: sfdfd
Age: 25
Employee ID: 11
Company Name: eeee
Qualification: mba
Salary: 60000
Subject: ob
Department: mba
Teacher ID: 111

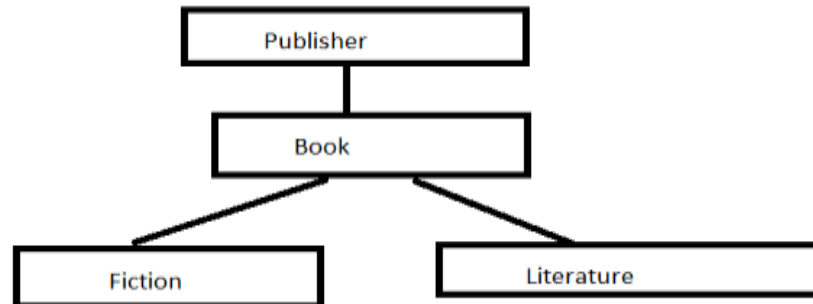
Details of Teachers:
Name: ammu
Gender: f
Address: qqweq
Age: 34

```

Employee ID: 10
Company Name: icet
Qualification: mca
Salary: 50000.0
Subject: cn
Department: mca
Teacher ID: 110

Name: feby
Gender: m
Address: sdfd
Age: 25
Employee ID: 11
Company Name: eeee
Qualification: mba
Salary: 60000.0
Subject: ob
Department: mba
Teacher ID: 111 */

4. CO3.4 Write a program that has class **Publisher**, **Book**, **Literature** & **Fiction**. Read the information and print the details of books from either the category, using **hierarchical inheritance**. //BookInheritance.java



```
import java.util.Scanner;
class Publisher
{
    String name;
    public void getPublisherDetails()
    {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter Publisher Name: ");
        name = sc.nextLine();
    }
}
class Book extends Publisher
{
    String title;
    int year;
    public void getBookDetails()
    {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter Book Title: ");
        title = sc.nextLine();
        System.out.print("Enter Publication Year: ");
        year = sc.nextInt();
        sc.nextLine(); // Consume newline
    }
    public void displayBookDetails()
    {
        System.out.println("\nPublisher: " + name);
        System.out.println("Title: " + title);
        System.out.println("Year of Publication: " + year);
    }
}
class Literature extends Book
{
```

```

String gen;
public void getLiteratureDetails()
{
    Scanner sc = new Scanner(System.in);
    System.out.print("Enter Generation: ");
    gen = sc.nextLine();
}
@Override
public void displayBookDetails()
{
    super.displayBookDetails();
    System.out.println("Generation: " + gen);
}
}
class Fiction extends Book
{
    String theme;
    public void getFictionDetails()
    {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter Theme: ");
        theme = sc.nextLine();
    }
    @Override
    public void displayBookDetails()
    {
        super.displayBookDetails();
        System.out.println("Theme: " + theme);
    }
}
public class BookInheritance
{
    public static void main(String[] args)
    {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter the number of literature books: ");
        int numLitBooks = sc.nextInt();
        Literature[] litBooks = new Literature[numLitBooks];
        System.out.println("Enter details for literature books:");
        for (int i = 0; i < numLitBooks; i++)
        {
            litBooks[i] = new Literature();
            litBooks[i].getPublisherDetails();
            litBooks[i].getBookDetails();
            litBooks[i].getLiteratureDetails();
        }
        System.out.print("Enter the number of fiction books: ");
        int numFicBooks = sc.nextInt();
    }
}

```

```

Fiction[] ficBooks = new Fiction[numFicBooks];
System.out.println("Enter details for fiction books:");
for (int i = 0; i < numFicBooks; i++)
{
    ficBooks[i] = new Fiction();
    ficBooks[i].getPublisherDetails();
    ficBooks[i].getBookDetails();
    ficBooks[i].getFictionDetails();
}
System.out.println("\nLiterature Books:");
for (Literature litBook : litBooks)
{
    litBook.displayBookDetails();
    System.out.println();
}
System.out.println("\nFiction Books:");
for (Fiction ficBook : ficBooks)
{
    ficBook.displayBookDetails();
    System.out.println();
}
}
}

```

/*E:\S2 JAVA LAB 2024\JavaRec>javac BookInheritance.java

E:\S2 JAVA LAB 2024\JavaRec>java BookInheritance

Enter the number of literature books: 2

Enter details for literature books:

Enter Publisher Name: Harper Lee, Sissy Spacek

Enter Book Title: To Kill a Mockingbird

Enter Publication Year: 2014

Enter Generation: 2

Enter Publisher Name: George Orwell and Christopher Hitchens

Enter Book Title: Animal Farm

Enter Publication Year: 2010

Enter Generation: 1

Enter the number of fiction books: 1

Enter details for fiction books:

Enter Publisher Name: Penguin Random House

Enter Book Title: Triple Romancing

Enter Publication Year: 2016

Enter Theme: novel

Literature Books:

Publisher Name: Harper Lee, Sissy Spacek

Book Title: To Kill a Mockingbird

Publication Year: 2014

Generation: 2

Publisher Name: George Orwell and Christopher Hitchens

Book Title: Animal Farm

Publication Year: 2010

Generation: 1

Fiction Books:

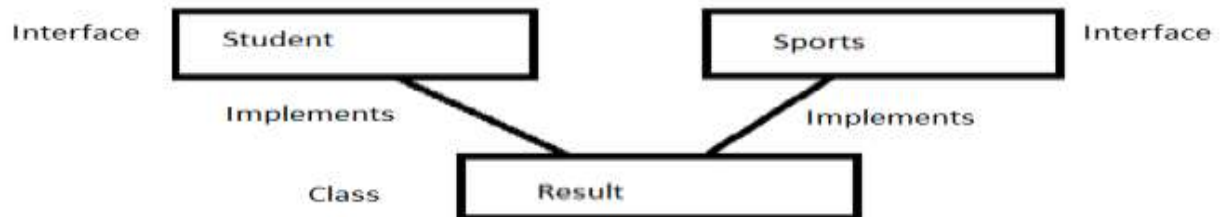
Publisher: Penguin Random House

Title: Triple Romancing

Year of Publication: 2016

Theme: novel */

5. **CO3.5 Create classes Student and Sports. Create another class Result inherited from Student and Sports. Display the academic and sports score of a student.**



```
import java.util.Scanner;
interface Student
{
    void displayScore();
}
interface Sports
{
    void displaySpScore();
}
class Result implements Student, Sports
{
    String name;
    private int acScore;
    private int spScore;
    public Result(String name, int acScore, int spScore)
    {
        this.name = name;
        this.acScore = acScore;
        this.spScore = spScore;
    }
    @Override
    public void displayScore()
    {
        System.out.println("Name: " + name);
        System.out.println("Academic score is: " + acScore);
    }
}
```

@Override

```
public void displaySpScore()
{
    System.out.println("Sports score is: " + spScore);
}
public class StudentSports
{
    public static void main(String[] args)
    {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter student's name: ");
        String name = sc.nextLine();
        System.out.print ("Enter academic score: ");
        int acScore = sc.nextInt();
        System.out.print ("Enter sports score: ");
        int spScore = sc.nextInt();
        Result r = new Result (name, acScore, spScore);
        r.displayScore ();
        r.displaySpScore ();
    }
}
```

/* E:\S2 JAVA LAB 2024\JavaRec>javac StudentSports.java

E:\S2 JAVA LAB 2024\JavaRec>java StudentSports

Enter student's name: ammu

Enter academic score: 69

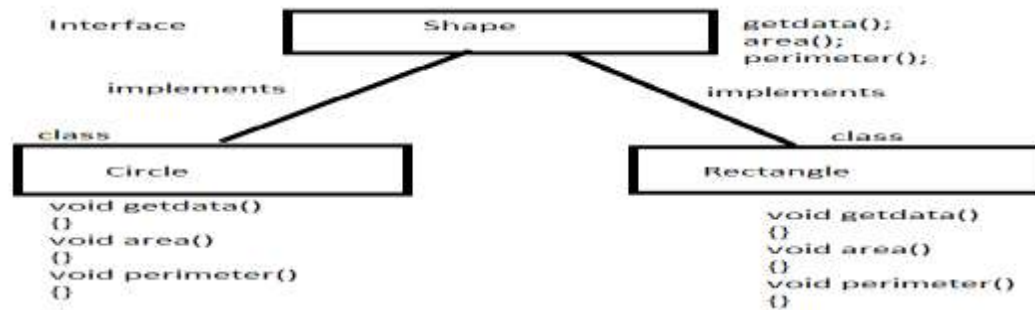
Enter sports score: 15

Name: ammu

Academic score is: 69

Sports score is: 15 */

/*6. CO3.6.Create an interface having prototypes of functions area() and perimeter(). Create two classes Circle and Rectangle which implements the above interface. Create a menu-driven program to find the area and perimeter of objects.
CO3Interface.java */



```

import java.util.Scanner;
interface Shape
{
    void getdata();
    void area();
    void perimeter();
}
class Circle implements Shape
{
    double pi = 3.14;
    double r;
    Scanner sc = new Scanner (System.in);
    @Override
    public void getdata()
    {
        System.out.println ("Enter the radius of the circle:");
        r = sc.nextDouble ();
    }
    @Override
    public void perimeter()
    {
        System.out.println("Perimeter of the circle: "+(2*pi*r));
    }
    @Override
    public void area()
    {
        System.out.println("Perimeter of the circle: "+(pi*r*r));
    }
}
class Rectangle implements Shape
{
    double l,b;
    Scanner sc = new Scanner(System.in);
  
```



```

@Override
public void getdata()
{
    System.out.println ("Enter the length of the rectangle:");
    l = sc.nextDouble ();
    System.out.println ("Enter the breadth of the rectangle:");
    b = sc.nextDouble ();
}
@Override
public void area()
{
    System.out.println("Perimeter of a rectangle: "+(l*b));
}
@Override
public void perimeter()
{
    System.out.println("Perimeter of a rectangle: "+(2*(l+b)));
}
}
public class CO3Interface
{
    public static void main(String[] args)
    {
        int ch;
        Scanner sc = new Scanner(System.in);
        Circle ob1 = new Circle();
        Rectangle ob2 = new Rectangle();
        do
        {
            System.out.println("\n1.Circle\n2.Rectangle\n3.exit");
            System.out.println("Enter your choice:");
            ch = sc.nextInt();
            switch(ch)
            {
                case 1: ob1.getdata();
                       ob1.area();
                       ob1.perimeter();
                       break;
                case 2: ob2.getdata();
                       ob2.area ();
                       ob2.perimeter ();
                       break;
                case 3: System.out.println("Exited...");
                       System.exit(0);
            }
        }while (true);
    }
}

```

```

/*E:\S2 JAVA LAB 2024\JavaRec>javac CO3Interface.java

```

```

E:\S2 JAVA LAB 2024\JavaRec>java CO3Interface

```

1. Circle
2. Rectangle
3. exit

Enter your choice:

1

Enter the radius of the circle:

2

Perimeter of the circle: 12.56

Perimeter of the circle: 12.56

1.Circle

2.Rectangle

3.exit

Enter your choice:

2

Enter the length of the rectangle:

2

Enter the breadth of the rectangle:

3

Perimeter of a rectangle: 6.0

Perimeter of a rectangle: 10.0

1.Circle

2.Rectangle

3.exit

Enter your choice:

3

Exited... */

8. Prepare bill with the given format using the calculate method from the interface.

Order No.

Date :

Product Id	Name	Quantity	unit price	Total
101	A	2	25	50
102	B	1	100	100
			Net. Amount	150

interface BillCalc

```
{  
    double calculate(double quantity, double unitPrice);  
}
```

class Bill implements BillCalc

```
{  
    private int orderNo;  
    private String date;  
    private String[][] items;  
    Bill(int orderNo, String date, String[][] items)  
    {  
        this.orderNo = orderNo;  
        this.date = date;  
        this.items = items;  
    }  
    @Override  
    public double calculate(double quantity, double unitPrice)  
    {  
        return quantity * unitPrice;  
    }  
    public void generateBill()  
    {  
        System.out.println("Order No.: " + orderNo);  
        System.out.println("Date: " + date);  
        System.out.println("Product Id\tName\tQuantity\tUnit Price\tTotal");  
        double netAmount = 0.0;  
        for (String[] item : items)  
        {  
            int productId = Integer.parseInt(item[0]);  
            String name = item[1];  
            double quantity = Double.parseDouble(item[2]);  
            double unitPrice = Double.parseDouble(item[3]);  
            double total = calculate(quantity, unitPrice);  
            netAmount += total;  
            System.out.println(productId+"\t" + name + "\t" + quantity + "\t" + unitPrice +  
                                "\t" + total);  
        }  
        System.out.println("\t\t\t\t\tNet. Amount\t" + netAmount);  
    }  
}
```

```

    }
}
public class CO3Bill
{
    public static void main(String[] args)
    {
        int orderNo = 123;
        String date = "2024-03-20";
        String[][] items = {{ "101", "A", "2", "25"}, {"102", "B", "1", "100"}};
        Bill bill = new Bill (orderNo, date, items);
        bill.generateBill ();
    }
}

```

/*E:\S2 JAVA LAB 2024\JavaRec>javac CO3Bill.java

E:\S2 JAVA LAB 2024\JavaRec>java CO3Bill

Order No.: 123

Date: 2024-03-20

Product Id	Name	Quantity	Unit Price	Total
101	A	2.0	25.0	50.0
102	B	1.0	100.0	100.0
Net. Amount			150.0	*/