



Data Structure and Algorithm (CSE-2001)

Assignment-II

Object Oriented Design

Question-1:

Define a class **Car** with two instance variables: *model* and *year*

Two member methods are:

- *setDetails()*: Set the details of the car.
- *displayDetails()*: Display the details of the car.

Now, create two objects of the Car class:

- Initialize the first object's values directly (by using the dot. operator) with model: "Toyota" and year: 2020.
- Accept the model and year through the keyboard input and set them to another object using the *setDetails()* method.

Now, display the details of both cars using the *displayDetails()* method.

Also, check which car is newer.

Question-2:

Define a class called **Complex** with instance variables *real*, *imag* and instance methods

- *void setData ()*
- *void display ()*
- *public Complex add (Complex, Complex)*

Write the main method in a separate class to implement the addition of two complex number with the given method signature as above.

Question-3:

Define a class named **Book** with the following member variables:

- *bookId* (an integer representing the unique ID of the book)
- *price* (a double representing the price of the book)
- *quantity* (an integer representing the quantity of books purchased)

Additionally, include a static variable *totalAmount* to keep track of the total amount spent on all books.

The class should have:

- A **parameterized constructor** to initialize the *bookId*, *price*, and *quantity*. **Note: Use "this".**
- A *displayDetails()* method to print the details of the book (i.e., *bookId*, *price*, *quantity*).
- A method to calculate and return the total cost for that book (*price * quantity*).
- Update the static variable *totalAmount* to include the total cost of the book purchased.

Now, write a separate **Main class** to simulate the purchasing of 5 different books. Display the details of each book, and finally, display the *total amount* spent on the purchase of all books.

Question-4:

Define a class **Deposit**. The instance variable of the class **Deposit** is mentioned below.

Instance Variable	Datatype
<i>Principal</i>	long
<i>Time</i>	integer
<i>Rate</i>	double
<i>Total_amt</i>	double

Initialize the instance variables *Principal*, *Time*, *rate* through **constructors**.

Constructors are overloaded with the following prototypes.

- **Constructor1:** *Deposit ()*
- **Constructor2:** *Deposit (long, int, double)*
- **Constructor3:** *Deposit (long, int)*
- **Constructor4:** *Deposit (long, double)*

Apart from constructor, the other instance methods are

- *display ()*: To display the value of instance variables
- *calc_amt()*: To calculate the total *amount*.

$$Total_amt = Principal + (Principal \times rate \times Time) / 100;$$

Question-5:

Create an abstract class **Shape** and the derived classes **Square**, **Triangle** and **Circle**. Write a java program to display area of different shapes.

Question-6:

Define a base class **Vehicle** with instance variables *model* and *year*. The instance variables are initialized through constructors. The prototype of the constructor is as below:

Vehicle (String, int)

Define a derived class **Car** with additional instance variables *carId* and *price*. The instance variables are initialized through constructors. The prototype of the constructor is as below:

Car (String, int, int, double)

Another instance method of the **Car** class is *carDisplay ()* to display the information of car details.

Write a program to:

- Create an object of the **Car** class and initialize its values.
- Display both the base class (**Vehicle**) and derived class (**Car**) details using the *carDisplay ()* method.

Question-7:

Define a base class **Shape** with an instance variable *color*. The instance variable is initialized through the constructor. The prototype of the constructor is as below: *Shape (String)*

Define two derived classes **Circle** and **Rectangle**, both inheriting from the **Shape** class.

Class Circle:

- Has an additional instance variable *radius*, initialized through the constructor.
- The constructor prototype is: *Circle (String, double)*
- A method *area ()* to calculate the area of the circle: $\pi * radius^2$.

Class Rectangle:

- Has two additional instance variables *length* and *width*, initialized through the constructor.
- The constructor prototype is: *Rectangle (String, double, double)*
- A method *area ()* to calculate the area of the rectangle: *length * width*.

Both the **Circle** and **Rectangle** classes should have a method *display ()* to display the shape's details, including color and the area of the respective shape.

Write a program to:

1. Create objects for both **Circle** and **Rectangle**.
2. Initialize their properties.
3. Display their details using the *display ()* method.

Question-8:

Define two interfaces, Flyable and Swimmable, with the following methods:

Interface **Flyable**:

- A method *fly ()* (e.g., "Flying not flying").

Interface **Swimmable**:

- A method *swim ()* (e.g., "Swimming or not swimming").

Define a class **Duck** that implements both **Flyable** and **Swimmable** interfaces. This class should:

- Implement the *fly ()* method from Flyable interface.
- Implement the *swim ()* method from Swimmable interface.
- Include an additional instance variable *name* (a String) to store the name of the duck, initialized through the constructor.
- A *displayDetails()* method to display the duck's name and its capabilities (whether it can fly or swim).

Write a program to:

- Create an object of Duck and initialize its name.
- Call the *fly()* and *swim()* methods using the Duck object.
- Display the duck's name and its abilities using the *displayDetails()* method.

Question 9:

Define a base class Employee with the following instance variables: *name*, *salary*

The Employee class should have:

- A constructor to initialize *name* and *salary*.
- A method *displayEmployeeInfo()* to display the details of the employee (i.e., *name* and *salary*).

Define a derived class Manager that extends the Employee class and adds an additional instance variable *department* (a String representing the department the manager manages).

The Manager class should:

- Have a constructor that uses the **super keyword** to call the Employee class's constructor to initialize name and salary.
- Include a *department* field, which should be initialized in the constructor.
- Implement a method *displayManagerDetails()* that calls the *displayEmployeeInfo()* method from the Employee class (**using super**) to display the employee's information and also displays the department of the manager.

Write a program to:

1. Create an object of **Manager** and initialize its *name*, *salary*, and *department*.
2. Call the *displayManagerDetails()* method to display the details of the manager.

Question-10:

Design a package that contains two classes **Student** & **Test**.

The **Student** class has data members as *name*, *roll* and instance methods

- *inputDetails()*
- *showDetails()*.

Similarly, the **Test** class has data members as *mark1*, *mark2* and instance methods

- *inputDetails()*
- *showDetails()*

Student is extended by Test. Another package carries interface **Sports** with 2 attributes *score1*, *score2*. Find grand total mark & score in another class

HOME ASSIGNMENT

Question-11:

Define a class called Student with instance variables *Roll*, *Name*, *DSA_Mark*.

Also, the class consists of instance methods

- *getdata()*: To provide input to the instance variable
- *showdata()*: To display the value of instance variable.

Write a program to create the details of 5 students. Display the information of the students who has secured the highest DSA_Mark.

Question-12:

Define a base class **Person** with instance variable *name*, *age*.

The instance variables are initialized through constructors. The prototype of constructor is as below.

Person (String, int)

Define a derived class **Employee** with instance variables *Eid*, *salary*.

The instance variables are initialized through constructors. The prototype of constructor is as below.

Employee (string, int, int, double).

Another instance method of Employee class is *empDisplay()* to display the information of employee details.

Question-13:

Create a class **Bank** with two instance variables:

bankNname, *depositAmount* and a class variable *totalAamount*.

Defined three member methods:

- *setBankName()* to set the bank name
- *setAmount()* to set the deposit amount
- *showData()* for displaying the bank name and deposit amount.

A person visited 5 different banks and deposit money. Create 5 array object of class **Bank** set the bank name and deposit amount (the minimum deposit amount is 1000) using member methods *setBankName()* and *setAmount()*. Call the *showData()* method to display the information with *totalAmount* deposited by the person. Also, design a method that displays the bank name where the person deposits the minimum amount.

Question-14:

Define a base class **Distance** with the following instance variables:

- *feet* (an integer representing the distance in feet)
- *inches* (an integer representing the distance in inches)

The **Distance** class should have:

- A constructor to initialize *feet* and *inches*.
- A method *displayDistance()* to display the distance in the format: *feet' inches"*.

Now, define a derived class **Measurement** that extends the **Distance** class and adds an additional instance variable *centimetres* (a double representing the distance in centimetres).

The **Measurement** class should:

- Have a constructor that uses the *super* keyword to call the **Distance** class's constructor to initialize *feet* and *inches*.
- Include a *centimetres* field, which should be initialized in the constructor.
- Implement a method *displayMeasurement()* that calls the *displayDistance()* method from the **Distance** class (using *super*) to display the distance in *feet* and *inches*, and also displays the distance in centimeters.

Question-15:

A point in the x-y plane is represented by its x-coordinate and y coordinate.

Design a class, **PointType** in Java, that can store and process a point in the x-y plane. You should then perform operations on the point, such as **showing the point, setting the coordinates of the point, printing the coordinates of the point, returning the x-coordinate, and returning the y- coordinate**. Every circle has a **centre** and a **radius**. Given the radius, we can determine the circle's area and circumference. Given the centre, we can determine its position in the x-y plane. The centre of a circle is a point in the x-y plane. Design a class, **CircleType** that can store the radius and centre of the circle. Because the center is a point in the x-y plane and you designed the class to capture the properties of a point from **PointType** class. You must derive the class **CircleType** from the class **PointType**. You should be able to perform the usual operations on a circle, such as setting the radius, printing the radius, calculating and printing the area and circumference, and carrying out the usual operations on the center.