Object Oriented Programming Lab

Paper Code: PC-CS592

Academic Year: 2023-24

List of Assignments

Learning OOP Concepts using Class, Object & Constructor

1. Write a Java program to create a class ***Discount*** containing SalesAmount as data member.

- SalesAmount should be given as input using appropriate input( ) method/ constructor defined in the class.
- In the class, define 2 methods viz. compute1( ) and compute2( ), those will compute the discount to be given using **if-else** construct and **ternary operator** respectively following the below mentioned conditions:
  - If SalesAmount is less than INR.10000/-, no discount given,
  - If SalesAmount is greater than INR.10000/- and less than INR.20000/-, 3% discount will be given,
  - If SalesAmount is greater than INR.20000/- and less than INR.40000/-, 5% discount will be given,
  - If SalesAmount is greater than INR.40000/- , 10% discount will be given.
- Add display( ) function to display the calculated discount.

Extra Assignment:

E1.1. Create a class named Complex containing data members a and b and member methods add(), sub(), Complex (int, int), display(). Create objects of Complex class and perform the addition & subtraction and display the result in each case.

[**Hint:** For addition of 2 Complex objects, use the formula: (a1+ib1) + (a2+ib2) = (a1+a2) + i(b1+b2)]

[N.B. Assume suitable return types of methods.]

Learning Overloading, Inheritance & Overriding

4. Create a class Arithmetic containing following data members

a : integer b: float c: char d : double.

and the following member functions

```
Arithmetic( int, float, char, double);   //  Parameterized constructor
void add(int, double);                   //  Polymorphic add function for adding to a and d
                                         //  data members of the Object
void add(int, float, double);            //  Polymorphic add function
void add(float, int, double);            //  Polymorphic add function
Arithmetic add(Arithmetic);              //  Polymorphic add function for adding 2 objects
void display( );                         //  Member function for showing the data members
```

Declare objects of the class Arithmetic as per requirement and invoke the member functions.

Comment on the performance of all the polymorphic add( ) functions.

Extra Assignment:

E4.1 Create a class Time containing the data members hr(int type), min(int type), sec(double type) and member functions

```
Time(int,int,double),    //  Parameterized constructor
showtime(),              //  Member function for showing the data members
add_time(int)            //  Polymorphic add function for adding min to that of Time object
add_time(int,int)        //  Polymorphic add function for adding hr & min to that of Time object
add_time(double)         //  Polymorphic add function for adding sec to that of Time object
add_time(Time)           //  Polymorphic add function for adding Time object to another Time object
```

Create objects of Time class and perform their addition.

5. Create a class named ***Student*** containing S_Name, Roll_No, Batch, Year_of_Adm, Stream as data members. Write different types of constructors for creating Student Objects such as

- Student(String, int);
- Student(String, int, String);
- Student(String, int, String, int);
- Student(String, int, String, int, String);

  Assume default values for the data members for which values have not been passed.
  Use showdata( ) method to display the data members for all objects so created.

2. Create a class ***Account*** containing data members Acc_No, C_Name, Contact_No as data members and a constructor for creating Account Objects and show( ) method for displaying the data members.
Create the following 3 sub-classes ,namely,

- ***Savings_Ac*** containing specialized features viz. ROI, Acc_Balance
- ***Current_Ac*** containing specialized features viz. Acc_Balance, Min_Balance
- ***TD_Ac*** containing specialized features viz. Principal, Term, ROI, Maturity_Balance

  Define suitable constructors in all of the above classes for creating objects and showdata( ) methods for displaying the data members. Also, define compute() method for different computations in all classes such as interest calculations etc.

3. Create a class ***ComputerProfessional*** containing EName, DutyHour and Prospect(Domain of values are "Excellent", "Good" and "Fair") as data members. Create 3 different child classes namely ***Developer, NetworkAdmin*** and ***DataOperator*** using ComputerProfessional as the super class. Include StudyHour as data member in Developer Class, PracticeHours as data member in NetworkAdmin Class, TypingSpeed as data member in DataOperator Class.
Create two more sub-classes of Developer Class namely ***JavaProfessional*** and ***PythonProfessional*** having ProficiencyLevel(Domain of values are "Beginner", "Intermediate" and "Pro") as specialized feature.
Use constructors to create the objects for all classes. Develop the appropriate display() methods in all of the classes to show the appropriate data members.

6. Write a Java program to create a class "***Shape***" and 3 other classes named ***Square, Rectangle*** and ***Triangle*** derived from it, all containing a overridden method cal_area( ) to calculate area of a Square or a Rectangle or a Triangle. Assume suitable data members(2 int type data members only) and member methods(get( ) and put( )) in all classes. Also validate the inputs.

Extra Assignment:

E4.1. Create a base class named ***Employee*** containing data members ECode, Ename, ContactNo, BasicSal, GrossSal, Dept and methods like computeTotSalary(), calcBonus(), showdata(). Different types of employees like ***Manager, Engineer*** and ***Clerk*** may have their implementations of the methods present in base class Employee. Each type of employee may have its logic in its class, e,g, if calcBonus() is present for a specific employee type, only that method would be invoked to provide hike in salary(by addition of Bonus). Use constructors in every class for creation of objects. Implement using suitable Java Code.
[N.B. 1.Assume suitable data-types of data members and return types of methods.
2.GrossSal=BasicSal+DA+HRA+MA where DA=75% of BasicSal, HRA=15% of BasicSal, MA=10% of BasicSal]

E6.1. Illustrate with a Java Program that in case of Multilevel Inheritance, overriding can be stopped after any level of inheritance.

E6.2. Illustrate with a Java Program that if the static method is redefined by a derived class, then it is not overriding, ***it is hiding***.

E6.3. Create an abstract class 'Bird' containing the following members:

Name(String),Colour(String)
Bird(String,String)
Fly();

Create a derived class named 'FlyingBird'containing no new data members except the over-ridden Fly() method .Also create another derived class neamed 'NonFlying_Bird' containing no new data member except the over-ridden Fly() method .Implement using a suitable java function.

Using Wrapper Class and Array

7. a) Convert the following primitive data types into corresponding wrapper objects
   - int, char, float, double.

   b) Store null values in the int and float variables.
   c) Store the Wrapper Objects into primitive types.
   d) Considering an array of characters, check whether individual characters are of lowercase, uppercase, a digit or a whitespace character.

8. Implement Bubble sort/ Insertion sort by taking user input inside an array of integers.[Note: Use Object Oriented features only]

9. Multiply two matrices by taking user input in two 2D Arrays of integers. Check whether the dimensions of matrices conform to the rule of Matrix Multiplication. [Note: Use Object Oriented features only]

Learning Interface

10. a) Create an interface named **Area** containing a final static variable, pi=3.14F and a method named cal_area (float dim1, float dim2).

    b) Create 2 classes named **Circle** and **Rectangle** to implement the Area interface (that is, define respective cal_area() method).

[**Hint:** - Use an interface object to refer to Circle object and Rectangle object to display the respective areas]

11. Create a class Student containing Name & Roll_No as data members. Create a sub-class **Test_marks** containing Marks1, Marks2 as data members. Create an Interface **CCA_Score** containing CCA_credit data member and show_cca_cred() method.

- Create a sub-class Sem_Result that inherits from Test_marks and implements CCA_Score interface.
- Define suitable get_data() and put_data() methods in different classes.
- Create objects of Sem_Result class to display all details of 2 students.

Using Java System Packages and User Defined Packages

12. Write a Java program to import Math Class defined within java.lang system package and use the value of PI defined there for finding the area and perimeter of a circle by accepting radius as input through suitable methods defined in a user defined Class named Math_operations.

Declare object of the class Math_operations and invoke the member functions as per requirement.

13. Modify Assignment11 as follows:

Design a package to contain the class **Student** and another package to contain the interface **CCA_Score**. Implement the *same problem* using these packages.

Explain the new design in your result discussion section.

14. Show with a suitable Java program that when we import any package, only the objects of public classes contained within the imported package can be created and used directly but the non-public classes contained within the package are hidden and cannot be used.

15. Write a Java program for adding a new class to an existing package. Also, implement two public classes in the Java program using packages.

16. Write a java program to show the following built-in-exceptions

1) ArithmeticException

2) ArrayIndexOutOfBoundsException

3) NullPointerException

4) NumberFormatException

Using Applets in JAVA

17. Create a java Applet to display your name at location (20,50) of a window.

18. Create and run applet to display the system date on your window.

19. Create A JAVA Applet that accepts 2 numbers from user and displays their sum.

Using Java Threads and Learn Multithreading

20. Write a Java program to create 20 threads and print their Ids.

[Hint: Inherit Thread Class]

21. Write a Java program to create 10 threads and print their names implementing Runnable interface. Use sleep() method to control working of threads. Use appropriate exception handling techniques.

Partha Pratim Saha
26/07/23 Mahua Nandi Pal

Signature of Faculties

26/7/23

Signature of HOD