Shri G. S. Institute of Technology and Science
Department Of Computer Engineering
CO24057: Object Oriented Programming System
Lab Assignment # 03

Submission Date: 3rd November 2021@23:59 (Wednesday)
Late Submission: Not allowed
No copying allowed. If found then students involved in copying will get zero marks in this assignment.

Q1. Write a Java Program to declare any class named Employee with the given attributes. Create another class which has a main method. Create at least 3 objects of the class employee, using constructor pass the user input values to assign to attributes.

1.1. Calculate salary using the rule:

Total salary is basic salary + HRA +Allowance
Note:
Dept: 1 for staff, 2 for accounts, 3 for admin
HRA=40% of basic salary
Allowance = 1500 for Dept 1
2500 for Dept 2
3500 for Dept 3

1.2. Print details of the object.

Q2. Create a java program based on the specifications given below:

Class Name: Creditcard

Data Members:
-String name,
-String cardNo,
-boolean enabled,
-int pin,
-String expiryMonth,
-int cardType, (Platinum =3, Gold =2, Silver =1),
-int currentCredit,
int creditLimit

Methods:
+CreditCard()
+changePin(int newPin)
+transact(int amt),
+changeCardStatus(boolean status)

+display()
Create a class CreditTester to include main method. Create objects of CreditCard in this main method and work on functionalities.

Note:

1. – indicates private access specifier, + indicates protected access specifier
2. transact() should check if the card is enabled + pin + credit limit.
3. Further based on the card type a discount is offered
   3% for platinum, 2% for Gold and 1% for Silver.

Q3. Design a class **Prism** to represent a prism, based on the following specifications:

Data Members: private double length(l), width(w), height(h)
Methods:
public void setPrism(): to assign values to l,w,h
public void double topArea(): returns top area of prism as l*w
public void double bottomArea(): returns bottom area of prism as l*w
public void double leftArea(): returns left area of prism as h*w
public void double rightArea(): returns right area of prism as h*w
public void double frontArea(): returns front area of prism as h*l
public void double backArea(): returns back area of prism as h*l
public void double bottomArea(): returns bottom area of prism as l*w
public double area(): returns a sum of the areas of all six sides as 2(l*w+h*w+h*l)

Develop a public main class TestPrism that tests all these methods.

Q4. Design a class **Fan** to represent a Fan, with the following specifications:
Private data members: String fanType, String manufacturer, String model, Boolean isOn
Public data members: enum Speed with 5 levels from 1 to 5
Methods:
public void setFan() and getFan() methods
public void on(): switch on the fan
public void off(): switch off the fan
public void speedUp(): to increase current speed, if not minimum 1
public void speedDown(): to reduce current speed, if not maximum 5

Develop a public main class TestFan that tests all these methods.

Q5. Create a java program to implement the following specifications:

A super class **Record** has to be defined to store the names and ranks of 50 students. Also define a *sub-class* **Rank** to find the highest rank along with the name. Details of both classes are given below:

Class Record:

Data Members:

name[ ] : to store the names of students

rnk[ ]: to store the ranks of students

Methods:

Record() : constructor to initialize data members

void readValues(): to store names and ranks

void display(): displays the names and the corresponding ranks

Class Rank:

Data members:

index: integer to store the index of the topmost rank

Methods:

Rank(): constructor to invoke the base class constructor and to initialize index=0

void highest(): finds the index/location of the topmost rank and stores it in index without sorting the array

void display(): displays the names and ranks along with the name having the topmost rank.

Specify the class Record giving the details of the constructor, void readValues(), and void display(). Using *the concept of inheritance*, specify the class Rank giving details of constructor, void highest() and void display(). Create a main class to test all the above methods.

Q6. Create a java program based on the following specifications:

A class Student defines the personal data of a student while another class Marks defines the registration number, name of subject and marks obtained by the student. The details of both the classes are given below:

Class name : Student

Data members

name : string to store name.

sex : string to store sex

age : integer to store age

Member functions

void inpdetails() : to accept values for data members.

void show() : to display personal data of student.

Class name : Marks

Data members

regnum : int to store registration number.

marks : int to store marks.
subject : String to store subject name.
Member functions
void inpdetails() : to accept values for data members.
void show() : to display exam and student details.

6.1. Specify the class **Student** giving details of the functions void inpdetails() and void show().
6.2. Specify the class **Marks** using the *concept of inheritance and method overriding and use of super keyword*, giving details of the functions void inpdetails() and void show(). Create a main class to test the methods.

Q7. Create a java program based on the following specifications:
A superclass **Worker** has to be defined to store the details of a worker. Also define a subclass **Wages** to compute the monthly wages for the worker. The details/specifications of both the classes are given below:

Class name: Worker
Data Members:
Name: to store the name of the worker
Basic: to store the basic pay in decimals
Member functions:
Worker (…): Parameterised constructor to assign values to the instance variables
void display (): display the worker's details

Class name: Wages
Data Members:
hrs: stores the hours worked
rate: stores rate per hour
wage: stores the overall wage of the worker
Member functions:
Wages (…): Parameterised constructor to assign values to the instance variables of both the classes
double overtime (): Calculates and returns the overtime amount as (hours*rate)
void display (): Calculates the wage using the formula wage = overtime amount + Basic pay and displays it along with the other details of the worker.

7.1.Specify the class **Worker** giving details of the constructor () and void display ( ).

7.2. Using the concepts of inheritance, specify the class **Wages** giving details of constructor ( ), double-overtime () and void display (). Create a main class to test the methods.

Q8. Create a java program based on the following specifications:

A line on a plane can be represented by coordinates of the two-end points p1 and p2 as p1(x1, y1) and p2(x2, y2).
A superclass **Plane** is defined to represent a line and a subclass **Circle** to find the length of the radius and the area of the circle by using the required data members of the superclass. Some of the members of both classes are given below:

Class name: Plane
Data members:
x1: to store the x-coordinate of the first endpoint
y1: to store the y-coordinate of the first endpoint
Member functions/methods:
Plane (int nx, int ny): parameterized constructor to assign the data members x1 = nx and y1 = ny
void show(): to display the coordinates

Class name: Circle
Data members:
x2: to store the x-coordinate of the second endpoint
y2: to store the y-coordinate of the second endpoint
radius: double variable to store the radius of the circle
area: double variable to store the area of the circle
Member functions/methods:
Circle(…): parameterized constructor to assign values to data members of both the classes
void findRadius(): to calculate the length of the radius using the formula:

$$\sqrt{(x1 - x2)^2 - (y1 - y2)^2}/2$$

assuming that x1, x2, y1, y2 are the coordinates of the two ends of the diameter of a circle
void findArea(): to find the area of a circle using the formula: πr^2. The value of pie(π) is 22/7 or 3.14
void show(): to display both the coordinates along with the length of the radius and area of the circle

8.1. Specify the class **Plane** giving details of the constructor and void show()
8.2. Using the concept of inheritance, specify the class Circle giving details of the constructor, void findRadius(), void find Area() and voidShow(). Create a main class to test the methods.

Q9. Create a java program based on the following specifications:
A superclass **Stock** has to be defined to store the details of the stock of a retail store. Also define a subclass **Purchase** to store the details of the items purchased with the new rate and update the stock. Some of the members of the classes are given below:

Class name: Stock
Data members/instance variables:

item: to store the name of the item
qt: to store the quantity of an item in stock
rate: to store the unit price of an item
amt: to store the net value of the item in stock
Member functions:
Stock (…): parameterized constructor to assign values to the data members
void display(): to display the stock details

Class name: Purchase
Data members/instance variables:
pqty: to store the purchased quantity
prate: to store the unit price of the purchased item
Member functions/ methods:
Purchase(…): parameterized constructor to assign values to the data members of both classes
void update (): to update stock by adding the previous quantity by the purchased quantity and replace the rate of the item if there is a difference in the purchase rate. Also, update the current stock value as (quantity * unit price)
void display(): to display the stock details before and after updation.

9.1. Specify the class Stock, giving details of the constructor() and void display().
9.2. Using the concept of inheritance, specify the class Purchase, giving details of the constructor(), void update() and void display(). Create a main class to test the methods.

Q10. Create a java program based on the following specifications:
A superclass Number is defined to calculate the factorial of a number. Define a subclass Series to find the sum of the series S = 1! + 2! + 3! + 4! + ……. + n!
The details of the members of both classes are given below:
Class name: Number
Data member/instance variable:
n: to store an integer number
Member functions/methods:
Number(int nn): parameterized constructor to initialize the data member n=nn
int factorial(int a): returns the factorial of a number
(factorial of n = 1 × 2 × 3 × …… × n)
void display()
Class name: Series
Data member/instance variable:
sum: to store the sum of the series
Member functions/methods:
Series(…) : parameterized constructor to initialize the data members of both the classes
void calsum(): calculates the sum of the given series
void display(): displays the data members of both the classes

10.1 Specify the superclass Number, giving details of the constructor, int factorial(int) and void display() methods.
10.2 Using the concept of inheritance, specify the class Series giving the details of the constructor(…), void calsum() and void display(). Create a main class to test the methods.