

Major Assignment # 2

Task # 1

Create a class Multiples. Assume that you want to generate a table of multiples of any given number. Write a program that allows the user to enter the number and then generates the table, formatting it into 10 columns and 20 lines. Interaction with the program should look like this (only the first two lines are shown)

Enter a number: 7

7	14	21	28	35	42	49	56	63	70
77	84	91	98	105	112	119	126	133	140

Task # 2

Create a class Temp. Write a temperature-conversion program that gives the user the option of converting Fahrenheit to Celsius or vice versa. Then carry out the conversion. Use floating point numbers:

```
Type 1 to convert Fahrenheit to Celsius,  
2 to convert Celsius to Fahrenheit: 1  
Enter temperature in Fahrenheit: 70  
In Celsius: 21.111111
```

Task # 3

Create a class calculator. Create an equivalent four function calculator. When it finishes calculation, the program should ask whether the user wants to do another calculation. The response can be 'y' or 'n'.

```
Enter first number, operator and second number: 10 / 3  
Answer = 3.333333  
Do another (y/n) ? y  
Enter first number, operator and second number: 12 + 100  
Answer = 112  
Do another (y/n) ? n
```

Task # 4

Create a class Fractions. Create an equivalent four function calculator for Fractions. When it finishes calculation, the program should ask whether the user wants to do another calculation. The response can be 'y' or 'n'.

```
Addition:          a/b + c/d = (a*d + b*c) / (b*d)  
Subtraction:        a/b - c/d = (a*d - b*c) / (b*d)
```

Multiplication: $a/b * c/d = (a*c) / (b*d)$
Division: $a/b + c/d = (a*d) / (b*c)$

The user should type the first fraction, an operator and a second fraction. The program should then display the result and ask whether the user wants to continue.

Task # 5

Create an employee class. The member data should comprise an int for storing the employee number and a float for storing the employee's salary. Member functions should allow the user to enter this data and display it. Write a main() that allows the user to enter data for three employees and display it.

Task # 6

Create a distance class. The member data should comprise an int for storing the distance provided, a string for choice of the measurement unit and another int for conversion. Member functions should allow the user to enter the distance in either of the three units (inches, millimeters, kilometers) and convert to the choice of the user and display it.

Task # 7

Create a class called tollbooth. The two data items are unsigned int to hold the total number of cars, and a type double to hold the total amount of money collected. A constructor initializes both of these to 0. A member function called payingCar() increments the car total and adds 0.50 to the cash total. Another function called noPayCar(), increments the car total but adds nothing to the cash total. Finally, a member function called display() displays the two totals. Make appropriate member functions.

The program should allow the user to push one key to count a paying car, and another to count a non-paying car. Pushing the Esc key should cause the program to print out the total cars and total cash and then exit.

Create a class called Student. It will contain member functions that will read and print student's detail using atleast 8 of the student's attributes for example: his/her ID, First name, Last Name, Contact, City, DOB, Current Semester, No. of courses enrolled etc.

The program should allow the user to enter his/her details and print them accordingly.

Task # 8.

Write a program to read time in seconds and convert in time format (HH:MM:SS) using classes and objects. The class must contain appropriate member functions and member variables.

The program should allow the user to enter seconds and print them in HH:MM:SS.

Example:

Enter time in seconds: 3666

The time is = 01:01:06

Task # 9

Raising a number n to a power p is the same as multiplying n by itself p times. Write as overloaded function `power()` having two versions for it. The first version takes double n and int p and returns a double value. Another version takes int n and int p returning int value. Use a default value of 2 for p in case p is omitted in the function call.

Task # 10

Write definitions for two versions of an overloaded function. This function's 1st version `sum()` takes an argument, int array, and returns the sum of all the elements of the passed array. The 2nd version of `sum()` takes two arguments, an int array and a character ('E' or 'O'). If the passed character is 'E', it returns the sum of even elements of the passed array and if the passed character is 'O', it returns the sum of odd elements. In case of any other character, it returns 0 (zero).

Task # 11

Create an inheritance hierarchy of Rodent: Mouse, Gerbil, Hamster, etc. In the base class, provide methods that are common to all Rodents, and override these in the derived classes to perform different behaviors depending on the specific type of Rodent. Create an array of Rodent, fill it with different specific types of Rodents, and call your base-class methods to see what happens.

Task # 12

Create a base class with two methods. In the first method, call the second method. Inherit a class and override the second method. Explain what happens.

Task # 13

Create a new class named Clock. Class Clock should have the following attributes:

Hour (Integer 1..12)

Minutes (Integer 0..59)

Seconds (Integer 0//59) isAM (boolean)

These attributes are represented by data fields with private access protection. Class Clock should support the following public extractor methods:

`getHours ()`

`getMinutes ()`

`getSeconds ()`

`getisAM ()`

An object of Class Clock may be constructed in any of the following ways:

-no parameters (sets clock to midnight)

-hour and isAM specified (sets minutes and seconds to zero) -all attributes specified

Class Clock should also support a setTime() method that sets all four attributes from argument values. Compile the new class.

Using these three classes, design a program that will accept dimensions of a triangle or a rectangle interactively and display area.

Task # 14

Consider classes:

CEO

Manager

Assistant Manager

Supervisor

Employees

Perform **Hierarchal Inheritance** which includes information that is relevant between CEO and all other classes. Each class must contain atleast 4 fields, a parameterized constructor for information input from the user and a DisplayInformation function.

Task # 15

Consider classes:

Head of Department

Co-ordinator

Faculties

Students

Perform **Multilevel Inheritance** in which information is passed on from the Head of Department to the next class. **You can assume any scenario as per your wish.** Each class must contain atleast 4 fields, a parameterized constructor for information input from the user and a DisplayInformation function.

Task # 16

Consider a scenario in which you have to construct a transcript of semesters that you have studied so far.

Make a class TranscriptSemester1 and furthermore till the semester you are currently studying in and provide the Semester GPA of each semester and Commulative GPA of all the semesters BY overriding relevant functions and displaying the information.

Task # 17

Imagine a publishing company that markets both book and audiocassette versions of its works. Create a class publication that stores the title (a string) and price (type float) of a publication. From this class derive two classes: book, which adds a page count (type int), and tape, which adds a playing time in minutes (type float). Each of these three classes should have a getdata() function to get its data from the user at the keyboard, and a putdata() function to display its data.

Write a main() program to test the book and tape classes by creating instances of them, asking the user to fill in data with getdata(), and then displaying the data with putdata().

Task # 18

Start with the publication, book, and tape classes. Add a base class sales that holds an array of three floats so that it can record the dollar sales of a particular publication for the last three months. Include a getdata() function to get three sales amounts from the user, and a putdata() function to display the sales figures. Alter the book and tape classes so they are derived from both publication and sales. An object of class book or tape should input and output sales data along with its other data.

Write a main() function to create a book object and a tape object and exercise their input/output capabilities.

Task # 19

Phoenix Systems is a well known computer hardware store located in L.A., California. The manager of the store wishes to develop a software through which he can store and view data about the hardware devices that he sells. The manager has hired a programmer to develop the software. The programmer has written the following class to store and view details about the devices:

```
class DeviceDetails {  
  
public:  
  
int deviceNo;  
  
string deviceName,deviceType;  
  
double devicePrice;  
  
  
DeviceDetails()  
{  
  
deviceNo=0;  
deviceName="";  
deviceType="";  

```

```

devicePrice=0.0;
}
DeviceDetails(int deviceNo, String deviceType){
deviceNo=deviceNo;
deviceType=deviceType;
}
void displayDetails(){
cout<<"Device number is "<< deviceNo;
cout<<"Device name is " << deviceNo;
cout<<"Device type is" << deviceNo;
cout<<"Device type is " << deviceNo;
cout<<"Device price is " << deviceNo;
};
Int main(){
DeviceDetails objDevice;
objDevice.displayDetails();
}

```

The program is giving compilation errors and not functioning as expected. Modify the program as follows:

- The user should be able to specify all details about a device at a time.
- The user should be able to specify only deviceNo and devicePrice when needed.
- The variables should not be accessible outside the class.
- The program should display all details of a device properly even when some details are not provided.

Task # 20

Create a class named Inventory. Your task is to maintain an inventory for a book store in which the following data has to be entered by the user:

- Book id
- Book name
- Book author
- Stock of the book

- Book's serial number
- Book's genre
- Number of pages

- 1) You need to initialize the values using a constructor and a display method to print the information.
- 2) A function name DuplicateRecord() must be formed which must stop the user from entering the same book twice.
- 3) A function named DuplicateAuthor() must be formed which counts the number of books in the inventory by the same author.
- 4) A menu should be formed allowing the user to choose from any of the above 3 mentioned conditions.

Task # 21

Create a class named ScientificCalculator. Your task is to construct a scientific calculator which contains the following functions

- CalculateSin()
- CalculateCos()
- CalculateTan()
- CalculateSumAvgMaxMin()

For the 4th function an array must be used.