National University of Computer and Emerging Sciences



Assignment 4

For

Object Oriented Programming Lab

|  |  |
| --- | --- |
| Course Instructor(s) | Mr. Usman Ghous |
| Lab Instructor(s) | Mr. Usman Ghous |
| Semester | Spring 2021 |

**FAST School of Computing**

# Instructions:

1. Make a word document with the naming convention “SECTION\_ LAB#\_ROLLNO” and put all your source code and snapshots of its output in it. Make sure your word file is formatted properly.
2. Plagiarism is strictly prohibited.
3. Do not discuss solutions with one another.

# Useful links

|  |
| --- |
| **Question#1** |

Write a class Employee having following attributes:  
1. String name  
2. Integer Age  
3. Float Salary  
Overload the appropriate operators for the following functionality:  
1. Input employee object  
2. Output employee object  
3. Convert the employee object to an integer (that integer should be the half of the employee’s  
salary)  
Write another class Person having the following attributes:  
1. String name  
2. Integer Age  
Convert an object of type Employee to person. You can use assignment operator for this purpose.

**Question#2**

Write C++ code for employees of a FAST-NUCES in which you calculate the salary of two  
same rank employees but their medical allowance and house rent have different % values.  
Write a class name *Employee* having data members:  
• String name  
• int Basic pay  
• int Medical Allowance  
• int House Rent  
• float Total\_Pay.

|  |  |  |
| --- | --- | --- |
| **Emp1** | | **Emp2** |
| Basic Pay | 20,000 | 20,000 |
| Medical Allowance % | 20 | 18 |
| House Rent % | 10 | 15 |

1.Task  
 Use constructor overloading for initializing the values.  
2. Task  
 Add the name of two employees (concatenate the name of two employee)  
3. Task

Calculate the total salary of each employee using operator overloading.  
4. Task  
 Compare the salaries of employees using comparison overloaded operator (==) and show  
a messages “Same Salary” if employee salary is equal if not then show a message  
“Different Salary”.  
5. Task  
 Use logical overloaded operators (! =,>, <) to compare salaries of **emp1** and **emp2**and  
Show a message employee’s “Not Equal”, “Greater” and “Less” respectively.  
6. Task  
 Employee operator++ (Overload pre-increment operator for salary).  
 Employee operator++ (int ) (Overload post-increment operator for salary).  
7. Task  
 Employee operator-- (Overload pre-decrement operator basic pay).  
 Employee operator-- (int ) (Overload post-decrement operator basic pay).  
8. Task  
 ostream& operator<<(ostream& os, const Complex& c);

Overload extraction operator so that it can display record in the format:  
• **Name:**

• **Basic Salary:**• **Medical Allowance:**• **Total Salary:**Define another class name STAFF having the following attributes:  
• String name  
• Salary  
Convert an object of type Employee to STAFF.

**Question#3**

Write a class **Complex** for complex numbers having the following data members:  
1. Float a  
2. Float b  
Write overloaded and default constructors for your class.  
Implement the following functionality for your class.

|  |  |
| --- | --- |
| float mag( ); | It will compute and return the magnitude of a complex number. The magnitude of a complex number a+bi is the quantity . |
| Complex add(Complex c); | The method accepts a complex number c, adds it with *this* complex number and returns the answer as another complex number. The addition of two complex numbers, a+bi and c+di is defined as follows:  (a+bi)+(c+di) = (a+b) + (c+d)i |
| Complex mul(Complex c); | The method accepts a complex number c, multiplies it with *this* complex number and returns the answer as another complex number. The multiplication of two complex numbers, a+bi and c+di is defined as follows:  (a+bi)\*(c+di) = (ac-bd) + (ad+bc)i |
| Complex operator++; | Overload pre-increment operator |
| Complex operator++(int); | Overload post-increment operator |
| Complex operator--; | Overload pre-decrement operator |
| Complex operator--(int); | Overload post-decrement operator |
| ostream& operator<<(ostream& os, const Complex& c); | Overload extraction operator so that it can display *this* complex number, in the format: **a+bi** |

**Question#4**

Let us assume you are going to print event cards of an upcoming event in your university.

Eventcard is a class and has the following private attributes:

 1. Event Name

 2. Date of event

 3. Time of event

 4. Venue (address of event)

 Use friend function to print all three attributes on console.

 The print function will be a function that is not a part of class itself but a friend of event

card class.

**Question#5**

* Make a class ***myclass*** and declare private member secret of type int.
* Initiliaze it with 0
* It has a member function print which prints the value of secret.
* Make ***secondclass*** its friend.
* Create class ***secondclass*** which has a method change and it changes the value of secret to

any user desired value. (You may take input of new value or set it by yourself, both of

these should be done from main).

* Now display the changed value by using print function of ***myclass***.

**Question#6**

Write a class Person, having following private data members:

1. Name
2. Data of birth (a constant data member)
3. Count (a static data member)
4. CNIC (a constant data member)

Count should keep track of how many person objects are created. Set the value of count before

any object is created. Display it at the end of the program.

Public member functions:

1. Constant member function to access the Date of Birth of each person (get DoB)
2. Constant member function to access the CNIC of each person (get CNIC)
3. Display function for Person record output (const)

Display a person record Mr. X with DoB 1st January 2001.

**Question#7**

Modify the problem 6 as follows:

Add Input function to the class.

Make all member functions private.

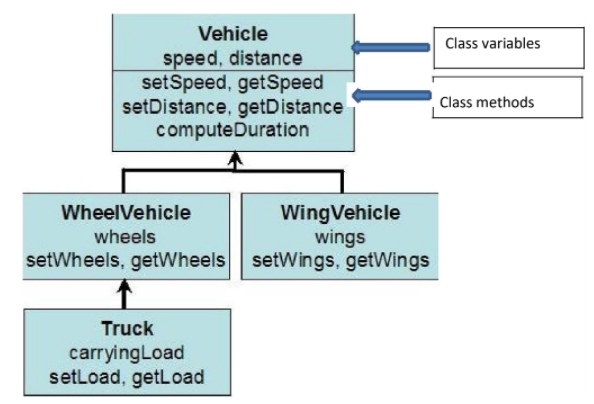
**Task:**

Write friend functions to access the member functions of the Person class separately (one friend

function for getting DoB, one for getting CNIC and one for displaying Person’s record).

**Question#8 (Classes, Inheritance, Friend functions)**

Design the following classes with given variables and methods using inheritance.



* **All four classes data members will be protected.**
* **Inheritance type is protected.**
* You have to write default constructors for each class. Child’s class constructor should

call parent class constructor.

* You have to write overloaded constructors for each class. Child’s class overloaded

constructor should call parent class overloaded constructor.

* You have to write **friend input** member function for each **leaf classes** to print the data

members to console. This function will input all the data member values by accessing

them. i.e. friend function for Truck class will be InputTruck(Truck T).

* You have to write **friend print** member function for each **leaf classes** to print the data members to console. This function will print all the data member values by accessing them.
* **Create objects of each leaf class and demonstrate your work with proper output.**