Object Oriented PROGRAMMING (CS217)

ASSIGNMENT- 6 & 7

Instructions

You are free to consult each other for verbal help. However **copying or sharing the code with each other will not only result into the cancellation of the current assignment, it may impact your grade in all the assignments and exams as well.**

Deadline: 20/05/2020

Question #01

Write a Person class having the following:

1. String Name
2. Blood group
3. Pure Virtual Display Blood group method

Inherit the following classes from Person having the same attributes and behavior of Person (polymorphism).

1. A
2. B
3. C

Task:

1. From main(), display blood group of each person on the console using Pointer of base class to the derived class.
2. Write destructor when you realize that this is the end of program.

Question #02

Finding the areas of Polygon family (Square+Rectangle +Triangle) using Polymorphism and Virtual function.

Define a base class Polygon having the following attributes:

1. Float length
2. Float width

Pure Virtual Member functions:

1. Float Area() to calculate Area
2. Float Perimeter() to calculate Perimeter
3. Void Display() to calculate Display Area and Perimeter

Define the derived classes and inherit above attributes and function publically:

1. Square
2. Rectangle
3. Triangle

Task

1. Write default constructor and Overloaded constructor to overload the values of Length and Width.

2. Calculate the Areas and Perimeters of Polygon family.

3. Show the results on the console using Display () function.

4. Write base class pointer "*Bptr" which points to the objects of derived classes and display the area and perimeter of subclass using Static and dynamic binding

Hint:

1. Static binding: Using. operator .
2. Dynamic Binding: Using -> operator

Hint:

1. Square Area/Perimeter=4*L
2. Rectangle A=2*(LxW) P=2*(L+W)
3. Triangle A= (LxW)/2 P= (L+W+H) .

Question #03

Write a program which can detect the sounds of Animals. Write a class Animals having the following attributes:

1. String Name
2. Sound()

Inherit the following classes from Animals having the same attributes and behavior of base class (polymorphism).

1. Cat
2. Dog
3. Tiger_Family
4. Deer

From Tiger_family class inherit the following sub classes publically:

1. Tiger
2. Lion
3. Leopard

Task:

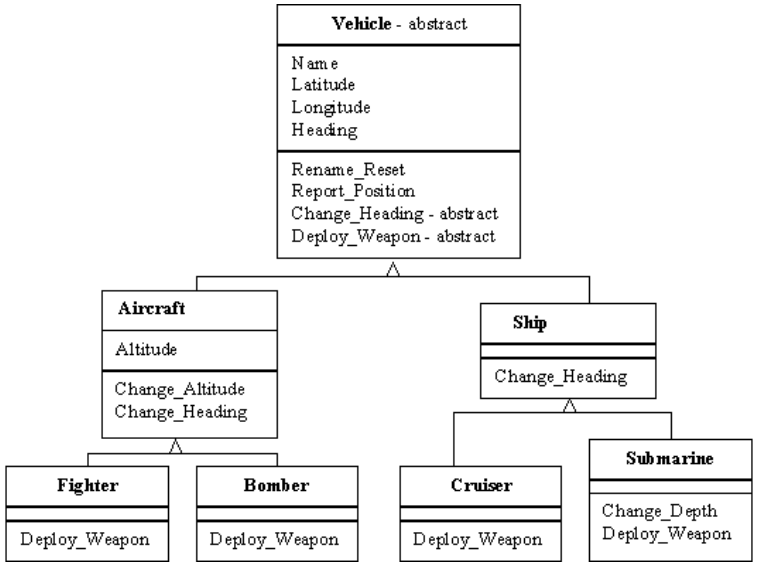
1. From main(), display sound of animals on the console using Pointer of base class to derived classes.

2. From main () also find the type of error and comment the error if a Derived class pointer is assigned a Base class address or pointer.

Hint: Like sound for Cat is "Meowmeow" :P

Question #04

Implement the following



Question #05

Create a class **BankAccount** having following attributes:

i. accountHolderName
ii. accountNo
iii. Balance

And following functions

i.	withdraw()	pure virtual
ii.	deposit()	pure virtual
iii.	checkBalance()	

Inherit 3 classes implementing the above interface.

SavingAccount, **CheckingAccount**, **InvestmentAccount**.

SavingAccount generates a profit of 5% on each deposit and deducts an amount of 10% if the customer withdraws amount in the last week of the month.

CheckingAccount doesn't involve any deduction or profit.

InvestmentAccount generates 10% profit on each deposit but there is no deduction rule defined so far.

Test your code by generating different bank accounts.

Note: (Abstract class can also be called as an interface and the derived classes providing the implementation of the pure virtual functions are known to be implementing that interface.)

Question #06

Create a class **Employee** having following attributes and functions

i. Name
ii. hireYear
i. monthlyPay() pure virtual
ii. annualPay() pure virtual

Inherit two classes from Employee class having mentioned attributes

SalariedEmployee	**HourlyEmployee**	
i. monthlySalary	i. hoursPerWeek	ii. hourlyWage

Create two classes **Manager** and **Staff** implementing the **SalariedEmployee class.**

Define two classes **FullTimeEmployee** and **PartTimeEmployee** implementing the **HourlyEmployee** class.

Question #07

Write a template class **Calculator** having following template functions

i. Add()
ii. Subtract()
iii. Multiply()
iv. Divide()
v. Modulus()

Test your code using different type inputs.

Note: Handle exceptions for string or character inputs.

Question #08

Implement **Stack** using Templates. Create a template class Stack and add following functions

i. Push()
ii. Pop()
iii. isEmpty()

Test your code with different data type inputs.

Note: You can implement using Arrays or Pointers, that's up to you. Add proper comments explaining your code to get marks.