AMITY UNIVERSITY

Object Oriented Programming

using

Programming Exercise 1(using class)

- Create the equivalent of a four-function calculator. The program should ask the user to enter a number, an operator, and another number. (Use floating point.) It should then carry out the specified arithmetical operation: adding, subtracting, multiplying, or dividing the two numbers. Use a switch statement to select the operation. Finally, display the result. When it finishes the calculation, the program should ask whether the user wants to do another calculation. The response can be ‘y’ or ‘n’.
- Some sample interaction with the program might look like this:
- Enter first number, operator, second number: 10 / 3
- Answer = 3.333333 Do another (y/n)? Y
- Enter first number, operator, second number: 12 + 100
- Answer = 112 Do another (y/n)? n

Computing area using classes

- Create a class area and three functions of class for computing area of square, rectangle and circle.
- Take the input from the user and decide which method should be called.

Programming Exercise 2

- Imagine a tollbooth at a bridge. Cars passing by the booth are expected to pay a 50 cent toll. Mostly they do, but sometimes a car goes by without paying. The tollbooth keeps track of the number of cars that have gone by, and of the total amount of money collected.
- Model this tollbooth with a class called tollBooth. The two data items are a type 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 total cars, number of cars that paid toll, not paid toll and cash collected.

Programming Exercise 3

- Create a class called time that has separate int member data for hours, minutes, and seconds.
- One constructor should initialize this data to 0, and another should initialize it to fixed values.
- Another member function should display it, in 11:59:59 format. The final member function should add two objects of type time passed as arguments.
- A main() program should create two initialized time objects (should they be const?) and one that isn't initialized. Then it should add the two initialized values together, leaving the result in the third time variable.

Programming Exercise 4

- Create a fraction class with data members num, den.
- Constructors – default ad parametrized
- Member data is the fraction's numerator and denominator. Member functions should accept input from the user in the form 3/5, and output the fraction's value in the same format.
- Member functions should add, subtract, multiply and divide two fraction values.
- Write a main() program that allows the user to repeatedly input two fractions and then displays the operation. After each operation, ask whether the user wants to continue.

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)`

Programming Exercise 5

- Write a class employee with name, id, salary scale and dept data members.
- ID should be automatically generated serially
- Constructors
- Functions:
  - Input employee details
  - Display employee details
  - Week off : Employee Id with even numbers have week off on Saturday
    Employee Id with odd numbers have week off on Sunday
  - Compute Salary:
    - If scale is high: basic + HRA + Da
      - HRA = 25% basic
      - DA = 120% Basic
    - If scale is medium
      - HRA = 15% basic
      - DA = 100% Basic
    - If scale is low
      salary is fixed to 12,000

Main function create an array of 10 employees.

Programming Exercise 6

- Write a program and input two integers in main and pass them to parametrized constructor of the class. Show the result of the addition of two numbers.
- Create a class highest and declare three data members of private type. Use constructor to initialize the data members. Write a program to find the biggest of three numbers using friend function

Programming Exercise 7

- Create a class circle with data member as radius.
- Constructors
- Functions to calculate area, perimeter, input , display, greater – Finds which circle has greater area (use the area function defined)
- Main function : Array of circles and find the circle with greatest area.

Programming Exercise 8

- Create a class text.
- Create a data member st of string type.
- The constructor of text will take string st as input.
- Create a member function wordcount which returns the count of the number of words in string st. Declare function wordcount as inline.
- Create a member function to reverse the words of string. For eg: “hello world”->”olleh dlrow”
- Use destructor to deallocate the space.

```
#include<iostream>
#include<string>
using namespace std;
class Text
{
    string st; //one dimensional string
    public:
    Text() //default constructor
    {
        cout<<"enter the string";
        getline(cin,this->st);
    }
    int wordcount()
    {
        int len=this->st.length(); // calculates the string length
        //this->st.at(len)='\0';
        return len;
    }
    void reverse()
    {
        cout<<"the original string is: "<<this->st<<endl;
        int l=wordcount();
        int c=0,a,b;
        for(int i=0;i<l+1;i++)
        {
            a=c;
            if(this->st[i]==' '|| this->st[i]=='\0')
           {
                b=i-1;
                c=i+1;
                while(a<b)
                {
                    char temp=this->st[a];
                    this->st[a]=this->st[b];
                    this->st[b]=temp;
                    a++;
                    b--;
                }
            }
        }
        cout<<"the reversed string is: "<<this->st<<endl;
    }
    ~Text()
    {
        int l=wordcount();
        this->st.erase(0,l); // delete the string
    }
};
int main()
{
    Text *t=new Text;
    t->reverse();
    delete t;
    return 0;
}
```

Complete the program

The ZooAnimal class definition below is missing a prototype for the Create function. It should have parameters so that a character string and three integer values (in that order) can be provided when it is called for a ZooAnimal object. Like the Destroy function, it should have return type void. Write an appropriate prototype for the ZooAnimal Create function.

```
class ZooAnimal
{
private:
char *name;
int cageNumber;
int weightDate;
int weight;
public:
void Destroy (); // destroy function
char* reptName ();
int daysSinceLastWeighed (int today);
};
```