

Objective Part

Compulsory

Q.No.1: Attempt all parts and each require answer 2 – 3 lines

(16*2=32)

1. Destructors (C++) "Destructor" functions are the inverse of constructor functions. They are called when objects are destroyed (de-allocated). Designate a function as a class's destructor by preceding the class name with a tilde (~).

2. Members of a class (variables and methods) marked with **public access specifier** are accessible to all members of the same class and all other classes. They can be referenced from any other class in any other directory also. Members of a class (variables and methods) marked with **protected access specifier** are accessible to only members of the same class and any other class that inherits it (i.e. its child class)

3. In object-oriented programming languages with multiple inheritances, the **diamond problem** (sometimes referred to as the "deadly diamond of death") is an ambiguity that arises when two classes B and C inherit from A, and class D inherits from both B and C. If D calls a method defined in A (and does not override the method), and B and C have overridden that method differently, then from which class does it inherit: B, or C?

4. Types of Inheritance are as follows:

- 1- Single Inheritance.
- 2- Multiple Inheritance.
- 3- Multilevel Inheritance.
- 4- Hybrid Inheritance (also known as Virtual Inheritance)

5. Operator overloading is an important concept in C++. It is a type of polymorphism in which an operator is overloaded to give user defined meaning to it. Overloaded operator is used to perform operation on user-defined data type. For example '+' operator can be overloaded to perform addition on various data types, like for Integer, String.

No, we can only overload the operators that exist. We cannot create new operators or rename existing operators. For example, we could not create an operator ** to do exponents.

6. When programs become more complex, it is often the case that different errors and things that would create problems should be handled (for example runtime errors). Purpose of **try and catch block**: Try block tries to solve the error but if error remains then throw block take that error and give it to the catch block. Catch block shows or display that error to the user.

8. Function Overriding: Function Overriding also allows to have two or more function with the same name but in this case, it is important to have Inheritance. When a function is already defined in the base class and we redefine it in derived class, that function is said to be overridden. In other words, if an inherited class contains a function with same name as that of its parent, then this concept is known as Function Overriding.

Function Overloading: Function overloading allows us to define two or more function having the same name but with different function body. It is useful in real world applications because practically, we encounter such situations very commonly.

9. Data encapsulation, sometimes referred to as data hiding, is the mechanism whereby the implementation details of a class are kept hidden from the user. The user can only perform a restricted set of operations on the hidden members of the class by executing special functions commonly called methods.

10. Operator "new" is used to make objects of the class.

12. A static variable is a single memory location associated with the class. A non-static variable (that is a member of a class) represents a different memory location for each instance of the class. A **static member** has only one copy of instance variables that share among all the objects of the class whereas a **non-static member** has its own copy of instance variable.

Subjective Part

Q2. Write a class Runner that contains the following data members:

- The name of runner
- The distance covered by runner

The class has following member functions:

- Get function to input runner name and distance
- Show function to display runner name and distance

Create two objects of the class and also display the name of runner who has covered longer distance.

```
#include<iostream>
#include<string.h>
#include<conio.h>
using namespace std;
class runner{
public:
    string name;
    long dist;
    runner(){ //Constructor
        name=" ";
        dist=0;
    }
    void get(){ //take input values
        cout<<"Enter Name and Distance:";
        cin>>name>>dist;
    }
    void display(){ //display output values
        cout<<"\n Name:"<<name;
        cout<<"\n Distance traveled:"<<dist;
    }
};
int main(){
    runner r1, r2;
    r1.get();
    r2.get();
    cout<<"This runner has covered longer distance:";
    if (r1.dist > r2.dist)
        r1.display();
    else
        r2.display();
    getch();
    return 0;
}
```



Visit

<https://tshahab.blogspot.com>
for more.

Q3. Write classes Array that contain the Array of integers as data members. The class contain the member functions:

- A constructor that initializes the Array element to -1
- Input function to input the values in Array
- Show function to display the values of Array
- A member function to make the values of Array two times

```
#include<iostream>
#include<conio.h>
```

```

using namespace std;
class array {
    private:
        int arry[10]; //The size of array isn't mentioned in question so we can make it 10
    public:
        array(){
            arry[10] = -1;
        }
        void input() { //take input values of array
            cout << "Enter values to array:" ;
            for(int x=0;x<=9;x++){
                cin >> arry[x];
            }
        }
        void show() { //Display value of array
            cout << "\nThe values of array is:" ;
            for(int y=0;y<=9;y++){
                cout << endl << arry[y];
            }
        }
        void twotimes(){ //Display value of array Two Time greater:
            cout << "\nThe values of array Two Time greater:" ;
            for(int z=0;z<=9;z++){
                cout << endl << 2*arry[z];
            }
        }
    };
    int main(){
        array a1;
        a1.input();
        a1.show();
        a1.twotimes();
        getch();
    }
}

```

Q4. Write a program that declares a shape class as abstract class. There are two classes (all sides are equal) and Right Triangle being derived from shape class. Write suitable program to calculate the area and perimeter of relevant shapes (i.e Right Triangle and square). Formulas used to calculate the area and perimeter are as follows:

$$\text{Area of square} = (\text{side})^2,$$

$$\text{Perimeter of square} = 4 * \text{side}$$

$$\text{Area of Right Triangle} = \frac{1}{2} * \text{base} * \text{height},$$

$$\text{Perimeter of Right Triangle} = \text{side1} + \text{side2} + \text{side3}$$

```

#include<iostream>
#include<conio.h>
using namespace std;
class shape { //abstrat class with pure virtual functions
protected:
    long ar, prmtr;
public:
    virtual void area()=0;
    virtual void parameter()=0;
};
class square : public shape {
private:
    long side; //all sides of the square are equal
public:
    square(){ //constructor
        ar=prmtr=side=0;
    }
    void get(){ //takes input from user
        cout << "Enter a side of square:" ;
        cin >> side;
    }
    void area(){

```

```

        ar=side*side;           //mentioned in question
        cout<<"Area of the square is:"<<ar;
    }
    void parameter(){
        prmtr=4*side;           //mentione in question
        cout<<endl<<"The perimeter of the square is:"<<prmtr;
    }
};

class triangle : public shape {
private:
    long base, height, hyp;      //three sides of right triangle
public:
    triangle(){}                //constructor
    ar=prmtr=base=height=hyp=0;
}
void input(){                  //takes input from user
    cout<<endl<<"Enter three sides of Right triangle:";
    cin>>base>>height>>hyp;
}
void area(){                  //mentioned in the question
    cout<<endl<<"The area of Right Triangle is:"<<ar;
}
void parameter(){             //mentioned in the question
    cout<<endl<<"The parimeter of Right Triangle id:"<<prmtr;
}
};

int main(){
    square s1;
    triangle t1;
    s1.get();
    s1.area();
    s1.parameter();
    t1.input();
    t1.area();
    t1.parameter();
    getch();
    return 0;
}

```

Visit
<https://tshahab.blogspot.com>
for more.

Q6. Create a car class that includes name as its data member. Create another class Toyota that is derived from class and includes his model # (int type) and color (string type) as his data member. Write appropriate constructors (multiple constructors) and display method to create and display objects of Toyota class.

```

#include<iostream>
#include<conio.h>
#include<string.h>
using namespace std;
class car {
protected:
    string name;
public:
    car(){}          //constrictor
    name=" ";
};

class toyota : public car {
private:
    string color;
    int model;
public:
    toyota(){}       //1st-constructor

```

Visit
<https://tshahab.blogspot.com>
for more.

```
        name="NULL";
        color="NULL";
        model=0;
    }
toyota(int){           //2nd-constructor
    name=" ";
    color=" ";
    model=0;
}

void display(){          //function to create and display details
    name="Toyota Corolla";
    color="Grey";
    model=2017;
    cout<<"Name of car:<<name<<endl<<"Model of Car:"<<model<<endl<<"Colour of the
    car:"<<color;
}
};

int main(){
toyota t1;
t1.display();
getch();
return 0;
}
```

