Roll no: 07

CE Ritesh Dahal , Theory Assignment I

1. What is object oriented programming (OOP)? Also explain the main features of OOP.

=) Object oriented programming (OOP) is a programming paradigm based upon objects (having both data and methods) that aims to incorporate the advantages of modularity and reusability. Objects are instances of classes, which are used to interact with another to design applications and computer programs.

The main features of OOP are

(i) The bottom-up approach in program design.
(ii) programs organized around objects, grouped in classes.
(iii) Focus on data with methods to operate upon object's data.
(iv) interaction between objects through functions.
(v) Reusability of design through the creation of new classes by adding features to existing classes.

2. How do structures in C and C++ differ? Illustrate with an example.

=> The differences of structures in C and C++ are:

i) In c we define struct keyword is necessary to create the structure type value while in c++ it is not necessary.

ii) There is no concept of access modifier inside the structure in c while in c++ we can find the access modifier (eg: private and public). By default all are private.

iii) In c there is no function inside the structure while in c++ we can define function that can access the data members of structure directly.

iv) Structure in c can not have static members while in c++ structure can have static members.

Exampel static member.

C

```
Struct Age {
    Static int x;
};

int main()
{ return 0;
} /*
```

Error: expected specifier-qualifier-list before 'static'
51 static int x;

C++

```
struct Age {
    static int x;
};

int main()
{ return 0;
    }.
```

3. Differentiate between default constructor and parameterized destructor? ~~what type of constru~~ with suitable example.

Default constructor	parameterized constructor
i) A constructor that is automatically generated by the compiler in the absence of any programmer-defined constructors.	(i) A constructor that is created by the programmer with one or more parameters to initialize the instance variables of a class.
ii) Has no parameters.	ii) Has one or more parameters.
iii) If the programmer has not written a constructor, the default constructor is automatically called.	iv) programmer should write his own constructor when writing a parameterized constructor.

```
#include<iostream>
using namespace std;
class Number1 {
    int x,y;
public
  Number1() {          // Default constructor.
        x=0;
        y=0; }

Number1(int a, int b) {
        x=a;
        y=b;
  }
  int getdatax()
      { return x; }
int getdatay(){
        return y; }
};
  int main(){
    Number1 a(20,30)
 cout<<" value of x is "<<a.getdatax()<<"
        value of y is "<< a.getdatay()<<endl;

  }
```

4.) What is the purpose of constructor and destructor? What types of constructor are created by the compiler by default? Explain with an example.

=) In c++ each time an instance of a class is created the constructor method is called. constructor is a special member function of class and it is used to initialize the objects of its class. ~~It is treated~~ . constructor constructs the value of data member of a class. Initial values can be passed as arguments to the constructor function when the object is declared.

Destructors are used to destroy the objects that have been created by the constructor within the c++ program. Destructor names are same as the class name but they are preceded by a tilde(~). The destructor neither takes an argument nor returns any value and the compiler implicitly invokes upon the exit from the program for cleaning up storage that is no longer accessible.

```
#include<iostream>
using namespace std;
class number1{
    int x;
  public;
number1(){              // constructor
      x=0;
      cout<<" object is created"; }

~number1()      // Destructor
 { cout<<"object is Destructed"; }

int getnumber(int a){
          x=a;
          return x; }
int displaynum()
    { return x;
    }
};
  int main(){
     number1 num;
     num.getnumber(6);

  cout<<" The number passed is "<<num.
              displayNum()<<endl;
        return 0;

}
```

5). Define structure. How does it differ with union? Illustrate with an example.

=) Structure is a collection of variables of different data types under a single name.

Structure	union
i) Struct keyword is used to define structure.	i) Union keyword is used to define a union.
ii) Members donot share memory in a structure.	ii) Members share the memory space in a union.
iii) Any member can be retrieved at any time in a structure.	iii) only one member can be accessed at a time in a union.
iv) Stores different values for all the members.	iv) Stores same value for all the members.

Structure

```
struct Employee{
  char x;
  int y;
  float z;
} e1; // size
```

size of e1 = 1+2+4
= 7

Union

```
union Employee{
char x; // size 1 byte
int y; // size 2 byte
float z; size 4 byte
} e1; size of e1 = 4 byte.
```

size of e1 = 4 (maximum size of element)

6. what is the difference between array and pointer variables? In what way they are similar?

=) The difference between array and pointer variable:

•) An array is a collection of elements of similar data type whereas the pointer is a variable that stores the address of another variable.

•) An array size decides the number of variables it can store whereas, a pointer variable can store the address of only one variable in it.

• Arrays are static in nature which means once the size of the array is declared, it cannot be resized according to users requirement. Where as pointers are dynamic in nature, which means the memory allocated can be resized later at any point in time.

• Arrays are allocated at compile time while pointers are allocated at runtime.

we can declare pointer variable and set it to point to the beginning of an array.

7. what do you mean by function? Illustrate the difference between pass by value and pass by reference.

=) A function is a self contained block of codes or sub programs with a set of statements that perform some specific task or coherent task when it is called.

Difference between pass by value and pass by reference are:

- In the call by value, copy of the actual argument is passed to the formal argument and the operation is done on formal argument.

- when the function is called by 'call by value' method, it doesn't affect content of the actual argument.

Pass by reference:

- Instead of passing the value of variable, address or reference is passed and the function operate on the address of the variable rather than value.

Here formal argument is alter to the actual argument, it means formal arguments calls the actual arguments.

8. Define operator overloading. write a program that illustrates the concept of unary operator overloading and binary operator overloading. Also state the use of friend function in case of binary operator overloading.

=) in c++, we can make operators to work for user defined classes. This means c++ has the ability to provide the operators with a special meaning for a data type, this ability is known as operator overloading.

```
#include <iostream>
using namespace std;

class Number {
    int a,b;
    public:
  Number () {
     int a=0
     int b=0;
  }
```

```
Number ( int c, ind d) {
        a=c;
        b=d;
 }
 void getNumber () {
    Cout << " Enter Two Number" << "\n";
     cin>> a.>> b;
     Number (a, b);
 }

 void operator ++ () {
    cout << " Increment " << ++a << "\n";
    cout << " Increment " << ++b << "\n";
 }

Number operator + (Number &n2) {
    Number N;
   N.a = a+n2.a;
   N.b = b+n2.b;
    return N;

};
 void display () {
    cout << " After adding : " << a << "\n";
    cout << " After adding: " << b << "\n";
  }
 };
```

```
int main(){
    Number n1, n2, n3;
    n1.getNumber();
    n1.operator++();
    n2.getNumber();
    n3= n1+n2;
    n3.display();
```

3.

output

Enter Two Number : 1, 2

Increment 2

11 3

Enter Two Number, 3, 6

After Adding : 5

After Adding : 9.

Friend function in binary operator overloading helps to access the private object of different class to do assign task.

9 When will you make a function inline? Why? In what ways is inline function advantageous.

=) we will make a function inline when the functions are small that called often. Inline functions run a little faster than the normal functions as the compiler replaces the function call statement with the function code itself and then compiles the entire code.

Advantages of inline function are

(i) function call overhead doesn't occur.

ii) It also saves the overhead of push/pop variables on the stack when function is called.

iii) It also saves overhead of a return call from a function.

10. what are static members and static functions? explain with appropriate syntax.

=) Static members are the special type of member of the class which can accessed through the member functions, but the function should be static member function.

A static member function is a special member function, which is used to access only static data members, any other normal data member cannot be accessed through static member function.

Syntax

```
#include <iostream>
using namespace std;
  class Demo
    { static int x; //static data member
      public
       static void print() // static function
              cout << " value of x; " << x;
    }  int Demo::x=10; static data member initi
                                    2ati
  int main()
    { Demo o;
       o.print();
```

11.) what is the purpose of the copy constructor? when is a copy constructor invoked? Explain with an example.

=) The purpose of the copy constructor are is to initialize one object from another of the same type; to copy an object to pass it as an argument to a function. to copy an object to return it from a function.

copy constructor is invoked in following cases:

1) when an object of the class is returned by value.

2) when an object of the class is passed (to a function) by value as an argument.

3) when an object is constructed based on another object of the same class.

4) when compiler generates a temporary object.

```
#include <iostream>
using namespace std;

class point {
        int x, y;
  public: ~~int~~
    point(int x1, int y1)
    { x=x1;
      y=y1; }

  point (const point &p2)
      { x=p2.x;
        y=p2.y; }
   int getx()
     { return x; }
   int gety()
     { return y; }
};
```

```
int main()
{
    Point p1(10, 15); // Normal constructor is c
    Point p2=p1; // copy constructor is called

cout << "p1.x =" << p1.getX() << ", p1.y ="
                                << p1.getY()

cout << "\n p2.x =" << p2.getX() << ", p2.y =
                                << p2.getY();

    return 0;
```

output;

p1.x = 10, p1.y = 15
p2.x = 10, p2.y = 15

12. what is constructor? Is it mandatory to use constructors in a class? why? what are the types of constructor?

=> A constructor in C++ is a special method that is automatically called when an object of a class is created.

If a class is not required to initialize its data member or doesn't contain data membe

there is no need write empty constructor explicitly on class object creation, default constructor implicitly called will be enough but if we want to overload ctt construction in the class default constructor will not be available and we need to construct other constructor.

Types of constructor:

i) Default constructor
ii) parameterized constructor
iii) copy constructor.

13. what is Inheritance? why is it important? Briefly describe the different type of Inheritance.

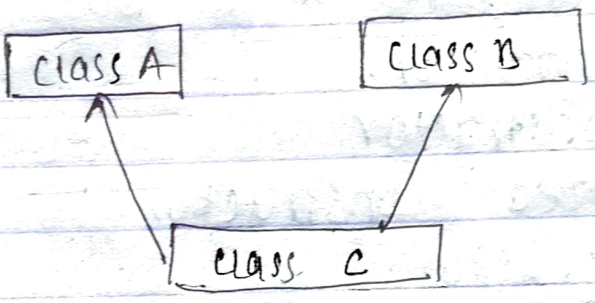
=) Inheritance is a mechanism where you can derive a class from another class for a hierarchy of classes that share a set of attributes and methods.

The main advantages of inheritance are code reusability and readability. when child class inherits the properties and functionality of parent class. we need not to write the same code again in child class. This makes it easier to reuse the code.

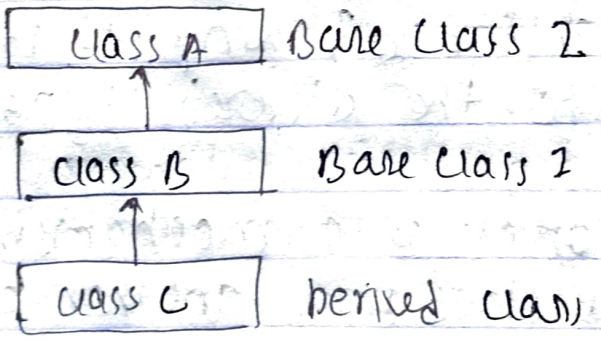
Types of inheritances are.

1) Single inheritance.

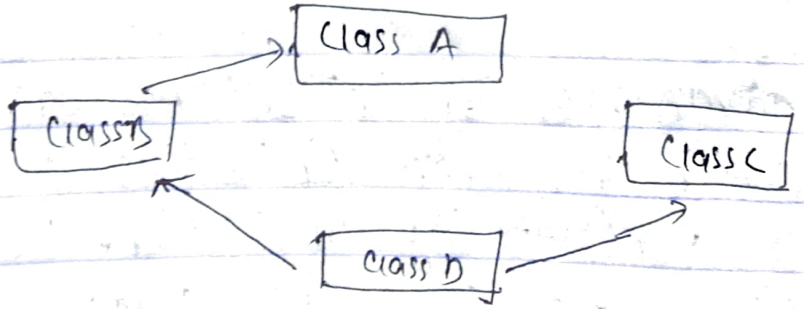
ii) Multiple Inheritance.



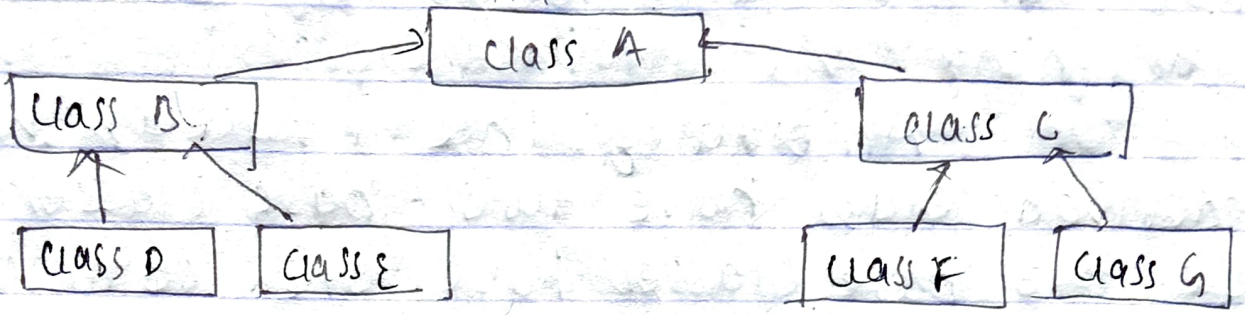
iii) multilevel inheritance



iv) Hybrid Inheritance



5) Hierarchical inheritance



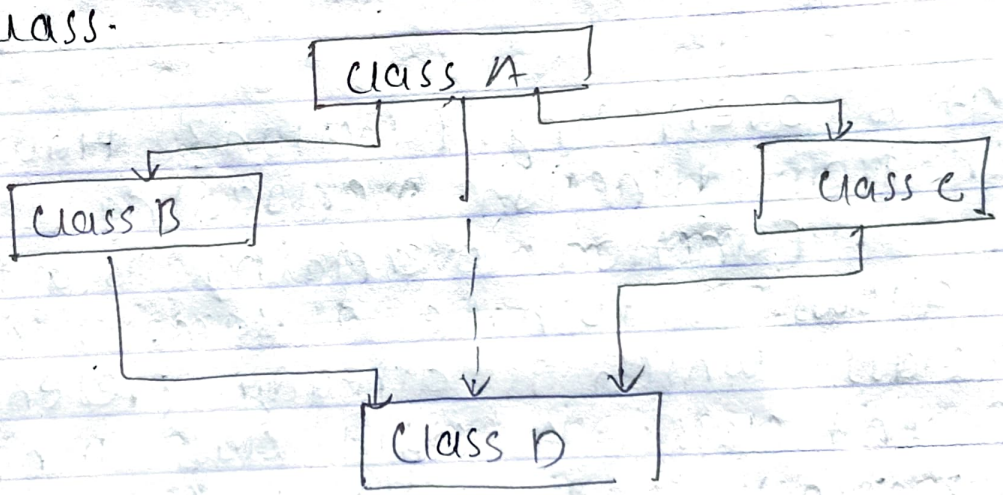
34. What is overriding function? How does it differ from function overloading?

=) Function overriding is a feature that allows us to have a same function in child class which is already present in the parent class. A child class inherits the data members and member function of parent class, but when you want to override a functionality in the child class then you can use function overriding. It is like creating a new version of an old function, in the child class.

Function overloading is when multiple function with same name exist in a class. function overriding is when function have same prototype in base class as well as derived class. function overloading can occur without inheritance. function overriding occur when one class is inherited from another class.

15. what sort of ambiguity can be resolved by defining virtual base classes?

=) Ambiguity in c++ occur when a derived class have two base classes and these two base classes have one common base class.



To remove multiple copies of class A from class D, we must inherit class A in class B and class C as virtual class.

6. what is friend function? what are merits and demerits of using friend functions?

⇒ A friend function, that is a "friend" of a given class, is a function that is given the same access as methods to private and protected data.

Merits of friend function

i) while defining the friend function, there is no need to use the scope resolution operator as friend keyword.

ii) The friend can be defined anywhere in the program similar to normal function.

iii) A function may be friend of more than one class.

Demerits

i) The friend function is declared in the class but it not a member function of class. To access the private data members of class it is necessary to create objects.

ii) when a function is friend of more than one class then forward declaration of class is needed.

17. what is function overloading and operator overloading? what is the difference between declaration and definition of a function? Explain with examples.

2) Function overloading in a c++ programming feature that allows us to have more than one function having same name but different parameter list.

operator overloading is the ability to provide the operators with a special meaning for a data type, ~~this also~~

The main difference between Function Declaration and function Definition in C++ is that function declaration indicates what the function is and function Definition indicates what the function does.

18. what do you mean by function overloading? when do we use this concept?

=> Function overloading in a C++ programming features that allows us to have more than one function having same name but different parameter list.

we use this concept of function overloading is to improve the code readability and allows code reusability.

9). When a class member is defined outside the class, which operator can be used to associate the function definition to a particular class? define that operator with a suitable example.

=) If the class member is defined or outside the class then we have to use scope resolution(::) operator along with class name along with function name.

```
#include<iostream>
  class
    using namespace std;
class cube
    {
      public:
      int side;
      int get volume();
    };
int cube:: get volume()
    {
      return side*side*side;
    }
```

```
int main()
{
    cube c1;
    c1.side=4;
    cout<<"volume of cube c1"<<c1.getvolume
                                ();
}
```

output

volume of cube c1 = 32.