OOP Lab Answers

Overloading vs Overriding :

* Declaring more than one function with the same name but with different parameters where as Declaring a function in the base class and derived class with the same name and parameter.
* One is compile time polymorphism where as the other is run-time polymorphism.
* Inheritance is involved in overriding.

Abstract class :

A class containing at-least one pure virtual function. Designed to act as a base class.

Syntax: class A{

Public:

Virtual void show()=0;

};

Multiple Inheritance:

Inheritance in which a class in derived from multiple base classes.

Class A{};

Class B{};

Class C: public A, public B{};

Class vs Object:

* A class is a group of data members and member functions used to hide the abstract details from the user whereas an object is an instance of the class.
* Declared using keyword class whereas an object is declared or created using the respective class name.
* Memory is not allocated at the time of declaration of a class where as memory is allocated for the creation of an object.
* Multiple object can be created for the same class.

Public vs Protected variable:

* Accessible throughout the program whereas not accessible throughout the program like protected members but can be inherited by the class derived from the base class like public members.
* Used while declaring member functions in a class so that they are accessible in the main method whereas protected is used in the base class for declaring data members to be inherited by the derived/child class.

Constructor:

Special type of function/method which is used to assign values directly to the object created which can be zero constructor/parameterised/copy.

Syntax of copy constructor:

className(className &var);

Initialization of static variable:

Int className :: var=0;

Use of super keyword:

Used to call the base-Class/ member functions of the base class

This pointer:

Also known as deference operator which allows us to get the value stored at the address held by the pointer.

Types of Inheritance:

* Single Inheritance: One class derived from a single base class
* Multiple Inheritance: One class derived from multiple base classes
* Multilevel Inheritance:One class derived from its parent class which is derived from another base class/ grand parent
* Hierarchical Inheritance: Two or more classes derived from a single class
* Multi -Path Inheritance: One class derived from two base classes which are derived from another single base class.

Use of friend function:

* To access all the non public members of the class.
* Used as a bridge between two classes by operating object of 2 different classes.
* Increase versatility of the program and enhances encapsulation.

Inline function over non-inline function:

* An enhancement feature in C++ that improves the execution time and speed of the program.
* Function call overhead doesn’t occur, There is no concept of control going from the function prototype to the function declaration, the lines of codes are just replaced.
* This eliminates call-linkage overhead and can expose significant optimization opportunities.

Constructor chain:

In a class, if a constructor tries to call another constructor with respect to the current object, the process is know as constructor chaining

Constructor overloading:

Declaring multiple constructors of the same name, but with different number and type of argument inside a class is known as constructor overloading.

Constructors are methods invoked when objects are created and have the same name as the class name. It falls under compile time polymorphism.

Order of constructor calling in multiple and multilevel:

Base Class 1-> Base Class 2-> Child Class

Grand Parent->Parent->Child

Virtual Function:

A member function in the base class which is expected to be redefined/overridden in the derived class. This is done when a pointer of the base class points to an object of the derived class.

A pure virtual function in a virtual function in C++ which is used when we need not to write any function definition and we only have to declare it. It is declared by assigning 0 in the declaration.

Use of Abstract class in C++:

Abstract class in a class in C++ which contains at least one pure virtual function. It is designed specifically to be used as a base class which can be shared by multiple derived class.

Use of OOP:

OOP provides a clear structure for the programs. OOP makes the code easier to maintain, modify and debug. OOP makes it possible to create full reusable applications with less code and shorter development time.Object-oriented programming aims to implement real-world entities like inheritance, hiding, polymorphism, etc in programming.

C vs C++:

C is a structural programming language following top down approach. It does not support any OOP features like virtual functions , operator overloading, exception handling, function overloading. Data security is less.

C++ is a object oriented programming language following the bottom up approach . It supports all the OOP features such as polymorphism, virtual function, operator overloading, exception handling and function overloading. Data security is more.

Copy Constructor:

Copy constructor is called mainly when a new object is created from an existing object, as a copy of the existing object.

Template:

Template is a very powerful tool/feature which is used to create generic classes/functions in C++ The simple idea is to pass the data type as parameter so the we need not write the same code for different data types.

Syntax: template<typename T>

T myMax(T x,T y)

{

Return (x>y)?x:y;

}

Int main()

{

Cout<<myMax<int>(3,7)<<endl;

Cout<<myMax<char>(‘g’,’h’)<<endl;

}

Exception Handling:

Exception handling is a mechanism that separates code that detects and handles exceptional circumstances from the rest of your program.

Syntax of operator overloading:

returnType operator+(className &ob1) ->Binary +

{}

returnType operator+(datatype k) ->Binary+

{}

returnType operator-() -> Unary -

{}

returnType operator++(int) //int is used to avoid ambiguity between post increment and pre Increment as both have the same symbol (++)

{}

reuturnType operator++()

{}

ReturnType operator==(className ob1)

{}

Friend istream &operator>>(istream &input,A &ob);

Friend ostream &operator<<(ostream &output, A &ob);

Data Abstraction:

Data Abstraction means displaying the essential information and hiding the other unnecessary details from the user. It refers to providing only the essential information about the data to the outside world and hiding the background details and implementation.

Polymorphism:

Polymorphism is one of the characteristics of object oriented programming which is used to implement different methods/functions with the same name.

Compile time polymorphism vs Run time polymorphism:

The function call is made at compile time, the compiler deduces the object and decides which call to bind to the object whereas in run time the function call is resolved at run time. Faster execution vs slow execution. Less flexible vs more flexible. Inheritance is involved in run time polymorphism.

Default arguments:

A default value is a value provided in a function declaration that is automatically assigned by the compiler if the calling function doesn’t provide a value for the argument, in case any value is passed the default value is overridden.

Reference Operator:

Returns the pointer address of the variable. It is denoted by &.

Encapsulation:

Wrapping up of data and information in under a single unit. Example can be a class where all the data members are stored and member functions are wrapped up hiding it from the user. Data is stored in the memory allocated during the instanciation of the class I.e. when the object is created.

Operator Overloading:

C++ allows us to specify more than one definition for an operator in the same scope

Operators which cant be overloaded:

* Ternary : ?
* Dot .
* Scope resolution ::

Class vs Structure:

Members are private by default. Declared using keyword class. Used for data abstraction and inheritance. Objects are created on heap memory. Reference type.

Members are public by default. Declared using keyword struct. Used for simply grouping of data. Objects are created on stack memory.Value type.

Object Slicing:

When an object of a derived class is assigned to an instance of the base class, part of the information is lost/ sliced away which is known as data hiding.