Revision :2015 Course code:3134

Course Title: Object oriented programming through C++

Qn No.	Scoring Indicator	Split up score	Sub Total	Total
1.	• Keywords are those words whose meaning is already defined by Compiler. These keywords cannot be used as an identifier. • Eg. Main, class , struct, cin, cout etc, any two keywords	1 1	2	
2.	• An enumerated data type is a user defined type which provides a way for attaching names to number • **enum shape { circle,square,triangle}**	2	2	
3.	• A destructor, is used to destroy the objects that have been created by a constructor. • Syntax ~ classname() {}	1 1	2	
4.	• The class which inherits the properties of another class is called Derived or Child or Sub class • The class whose properties are inherited is called Base or Parent or Super class	1 1	2	10
5.	• Call to a function is resolved at run time • Virtual functions	1 1	2	

	PART B			
II 1.	• Switch statement • The switch statement allows us to execute one code block among many alternatives • Syntax switch (expression) { case constant1: // statements break; case constant2: // statements break; . . . default: // default statements } • The expression is evaluated once and compared with the values of each case label. • If there is a match, the corresponding statements after the matching label are executed • If there is no match, the default statements are executed.	Explanation 3 Marks+ Syntax 2 Marks	6	6
2.	C++ provides the following classes to perform output and input of characters to/from files: • **ofstream:** Stream class to write on files • **ifstream:** Stream class to read from files • **fstream:** Stream class to both read and write from/to files.	3 Marks +	6	6

	Opening a File A file must be opened before you can read from it or write to it. Either **ofstream** or **fstream** object may be used to open a file for writing. And ifstream object is used to open a file for reading purpose only **Syntax** **void open(const char *filename, ios::openmode mode);** Closing a File When a C++ program terminates it automatically flushes all the streams, release all the allocated memory and close all the opened files **Syntax** **void close();** Writing to a File Writing information to a file from your program using the stream insertion operator (<<). Use an **ofstream** or **fstream** object instead of the **cout** object. Reading from a File Reading information from a file into your program using the stream extraction operator (>>), use an **ifstream** or **fstream** object instead of the **cin** object.	3 Marks for Explanation		
II 3.	• The constructors that can take arguments are called parameterized constructors. • Using parameterized constructor we can initialize the various data elements of different objects with different values when they are created. Example:- class integer	Explanation 2 Marks+		6

	{ int m,n; public: integer(int x, int y); }; integer:: integer (int x, int y) { m=x;n=y; } The argument can be passed to the constructor by calling the constructor implicitly. integer int 1 = integer(0,100); // explicit call integer int 1(0,100); //implicite call	Example 4 Marks	6	
II 4.	A default argument is a value provided in a function declaration that is automatically assigned by the compiler if the caller of the function doesn't provide a value for the argument with a default value. Example: // A function with default arguments, it can be called with // 2 arguments or 3 arguments or 4 arguments. int sum(int x, int y, int z=0, int w=0) { return (x + y + z + w); } int main() { cout << sum(10, 15) << endl; cout << sum(10, 15, 25) << endl; cout << sum(10, 15, 25, 30) << endl; return 0; } Output: 25	Explanation 2 Marks+ Example 4 Marks	6	6

	50 80 *Any sample program with explanation			
II 5.	• Only existing operators can be overloaded. • Overloaded operators must have at least one operand that is of user defined operators • We cannot change basic meaning of an operator. • Overloaded operator must follow minimum characteristics that of original operator • When using binary operator overloading through member function, the left hand operand must be an object of relevant class • ::.., sizeof, ?:, pointer cannot be overloaded	Each point carries 1 Mark	6	6
II 6.	• A virtual function is a member function that is declared within a base class and redefined by a derived class. • To create virtual function, precede the function's declaration in the base class with the keyword virtual. • When a class containing virtual function is inherited, the derived class redefines the virtual function to suit its own needs. • A virtual function cannot be a static member since a virtual member is always a member of a particular object in a class rather than a member of the class as a whole. • A virtual function cannot have a constructor member function but it can have the destructor member function The general syntax of the virtual function declaration is: class use_defined_name{ private: public: virtual return_type function_name1 (arguments); virtual return_type function_name2(arguments); virtual return_type function_name3(arguments); ---------- };	6Marks	6	6

II 7.	• A **single try statement** can have multiple catch statements. Execution of particular catch block depends on the type of exception thrown by the throw keyword. If throw keyword send exception of integer type, catch block with integer parameter will get execute. The format of multiple catch statement: try { //try block } catch(data type1 arg) { //catch block1 } catch(data type2 arg) { //catch block2 } .... catch(data typeN arg) { //catch blockN } As soon as an exception is thrown, Compiler searches for appropriate matching catch block. In case no matching found, the program is terminated.	Explanation 3 Marks + Syntax 3 Marks	6	6
III a	**primitive data types** available in C++. • **Integer**: Keyword used for integer data types is **int**. Integers typically requires 4 bytes of memory space and ranges from -2147483648 to 2147483647.	Datatypes 6 Marks+		15

	• **Character**: Character data type is used for storing characters. Keyword used for character data type is **char**. Characters typically requires 1 byte of memory space and ranges from -128 to 127 or 0 to 255. • **Boolean**: Boolean data type is used for storing boolean or logical values. A boolean variable can store either *true* or *false*. Keyword used for boolean data type is **bool**. • **Floating Point**: Floating Point data type is used for storing single precision floating point values or decimal values. Keyword used for floating point data type is **float**. Float variables typically requires 4 byte of memory space. • **Double Floating Point**: Double Floating Point data type is used for storing double precision floating point values or decimal values. Keyword used for double floating point data type is **double**. Double variables typically requires 8 byte of memory space. • **void**: Void means without any value. void datatype represents a valueless entity. Void data type is used for those function which does not returns a value. • **Wide Character**: Wide character data type is also a character data type but this data type has size greater than the normal 8-bit datatype. Represented by **wchar_t**. It is generally 2 or 4 bytes long. **Datatype Qualifiers/Modifiers**: As the name implies, datatype modifiers are used with the built-in data types to modify the length of data that a particular data type can hold. Data type modifiers available in C++ are: • **Signed** • **Unsigned**	Qualifiers 3 Marks	9	

	• **Short** • **Long**			
III b	There are two types of memory management operators in C++ • New • Delete new operator: is used for dynamic memory allocation .This operator can be used to create object of anytype. Syntax: Pointer variable= new datatype; Here new is a keyword and variable of type data type Eg: int * a= new int; In the above example, the new operator allocates sufficient memory to hold the object of data type int and returns a pointer to its starting point. The pointer variable a holds the address of memory space allocated. delete operator: The delete operator is used for releasing memory space when the object is no longer needed. Syntax: delete pointervariable; eg: delete a; Here delete is a keyword and the pointer variable is the pointer that points to the object already created in the new. The programmer must take care not to free or delete a pointer variable that has already been deleted.	3Marks + 3 Marks= 6 Marks	6	
IV	#include <iostream>			

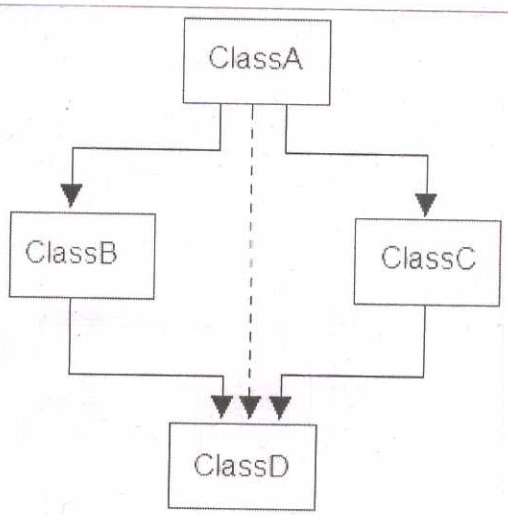
a	using namespace std; struct student { char name[50]; int roll; float marks; } s[10]; int main() { cout << "Enter information of students: " << endl; // storing information for(int i = 0; i < 10; ++i) { s[i].roll = i+1; cout << "For roll number" << s[i].roll << "," << endl; cout << "Enter name: "; cin >> s[i].name; cout << "Enter marks: "; cin >> s[i].marks; cout << endl; } cout << "Displaying Information: " << endl; // Displaying information for(int i = 0; i < 10; ++i) { cout << "\nRoll number: " << i+1 << endl; cout << "Name: " << s[i].name << endl; cout << "Marks: " << s[i].marks << endl; } return 0; }	Program 6 Marks	6	15

IV b	**Relational Operators :** We often compare two quantities and depending on their relation take certain decisions for that comparison we use relational operators. < is less than > is greater than <= is less than or equal to >= is greater than or equal to ==is equal to != is not equal to **Conditional Operator** A ternary operator /Conditional Operator requires two operands to operate Syntax: #include<iostream.h> void main() { int a, b,c; cout<<"Enter a and b values:"; cin>>a>>b; c=a>b?a:b; cout<<"largest of a and b is "<<c; } Enter a and b values:1 5 largest of a and b is 5 Bitwise Operators C ++ supports special operators known as bit wise operators for manipulation of data at bit level. They are not applied to float or double. operator meaning & Bitwise AND ^ Bitwise exclusive OR << left shift >> right shift	2+ 2+2= 6 Marks	6	

	~ one's complement			
V a	• A friend function is a function which is declared within a class and is defined outside the class. It does not require any scope resolution operator for defining . It can access private members of a class. • It is declared by using keyword "friend" #include<iostream> using namespace std; class B; //declare class B class A { private: int a; public: //constructor A(int a) { this->a = a; } friend int max(A a, B b); }; class B { private: int b; public: //constructor B(int b) { this->b = b; } friend int max(A a, B b);	Definitio n 2 Marks + Program 7 Marks	9	15

	}; int max(A a, B b) { return (a.a > b.b ? a.a : b.b); } int main() { A a(10); B b(15); cout << "Greatest is : " << max(a, b); return 0; }			
V b	• A constructor is a special member function whose task is to initialize the objects of its class. • The constructor is invoked whenever an object of its associated class is created. • It is called constructor because it constructs the values of data members of the class. Characteristics of constructors • They should be declared in the public section. • They are invoked automatically when the objects are created. • They do not have return type, not even void. • They cannot be inherited, though a derived class can call the base class constructor. • Like other c++ functions, they can have default arguments. • Constructors cannot be virtual. • We cannot refer to their addresses	3 Marks + 3Marks	6	
VI a	Call by value • Copy of the actual parameters is send to the function definition • Any change in the formal parameter will not affect actual parameter Call by reference	Explanation 4 marks +	9	15

	• Address o the actual parameter is send to the function definition • Any change in the formal parameter will affect actual parameter Sample program needed	Program 5 Marks		
VI b	• An inline function is a combination of macro & function. At the time of declaration or definition, function name is preceded by word inline. • When inline functions are used, the overhead of function call is eliminated. Instead, the executable statements of the function are copied at the place of each function call. • This is done by the compiler. • A function when defined as INLINE, the code from the function definition is directly copied into the code of the calling function. • Reduces space as no separate set of instructions in memory is written.	6 Marks		
VII a	• When a class is derived from a class which is also derived from another class, i.e. a class having more than one parent classes, such inheritance is called Multilevel Inheritance. • The level of inheritance can be extended to any number of level depending upon the relation. Sample program needed	Definitio n 3 Marks +Program 6 Marks	9	15
VII b	• A super class which may virtually transfer its public and protected members to a class that does not directly inherit from this base class • When two or more objects are derived from a common base class, we can prevent multiple copies of the base class being present in an object derived from those objects by declaring the base class as virtual when it is being inherited. Such a base class is known as virtual base class. • This can be achieved by preceding the base class' name with the word virtual.	Explanati on 4 Marks + Syntax 2 marks	6	



Syntax:

class derivedclassname: visibility mode virtual baseclassname
{
    Member declarations/definitions
};

VIII a	Binary operator means, an operator which works on two operands. For example, + is an binary operator, it takes single operand (c+d). So, when overloading an binary operator, it takes one argument (one is object itself and other one is passed argument). **Syntax for Binary Operator (Inside a class)** return-type operator operatorsymbol(argument) { //body of the function } **Syntax for Binary Operator definition (Outside a class)** return-type classname::operator operatorsymbol(argument) { //body of the function } **A sample program**	Explanation 3 Marks+ Sample program	9	15

		6Marks		
VIII b	These are also Called as Access Visibility Controls means they defined where a method and Data Member of class will be used either inside a class ,outside a class ,in inherited class or in main **Public Access**: - Specifies that data Members and Member Functions those are declared as public will be visible in entire class in which they are defined. Public Modifier is used when we wants to use the method any where either in the class or from outside the class **Protected Access**:- The Methods those are declared as Protected Access modifiers are Accessible to Only in the Sub Classes but not in the Main Program **Private Access**:- The Methods or variables those are declared as private Access modifiers are not would be not Accessed outside from the class or in inherited Class or the Subclass will not be able to use the Methods those are declared as Private they are Visible only in same class where they are declared. By default all the Data Members and Member Functions is Private, if we never specifies any Access Modifier in front of the Member and Data Members Functions.	3 *2=6 Marks	6	
IX a	Class template: The templates declared for classes are called class templates. A class template specifies how individual classes can be constructed similar to the normal class specification. These classes model a generic class which support similar operations for different data types. General Form of a Class Template template <class T> class class-name	Definitio n with syntax 3 Marks + Program		15

	{ .... }; The syntax for defining an object of a template class is: classname<type> objectname(arglist); Program for swapping #include<iostream.h> #include<conio.h> template <class T> class swap { T a,b; public: swap(T x,T y) { a=x; b=y; } void swapab() { T temp; temp=a; a=b; b=temp; } void showdata() { cout<<a<<b; } }; void main() {	6 marks	9	

```
int m,n;
float m1,n1;
cout<<"Enter integer values";
cin>>m>>n;
cout<<"Enter floating values";
cin>>m1>>n1;
swap<int> c1(m,n);
swap<float> c2(m1,n1);
c1.swapab();
c1.showdata();
c2.swapab();
c2.showdata();

}
```

X a	template <class T , class U> void multiply(T a, U b) { Cout<<"multiplication="<<a*b; } int main() { int a,b; float x,y; Cout<<"Enter two integer data"; Cin>>a>>b; Cout<<"Enter two float data"; Cin>>x,y; multiply(a,b);//multiply two integers multiply(x,y);//multiply two float multiply(a,x);//multiply integer and float return 0; }	9 marks	9	15
b	• Exceptions: Exceptions are runtime anomalies or unusual conditions that a program may encounter while executing • Two kinds of exceptions Synchronous exceptions- Errors such as "Out-of-range index" and "over flow" are synchronous exceptions Asynchronous exceptions- The errors that are generated by any event beyond the control of the program are called asynchronous exceptions	Definition 2 marks + Handling	6	

20

b

Method/Function Overloading and Overriding

	Overloading	Overriding
Definition	Methods having same name but each must have different number of parameters or parameters having different types & order.	Sub class have method with same name and exactly the same number and type of parameters and same return type as super class method.
Meaning	More than one method shares the same name in the class but having different signature.	Method of base class is re-defined in the derived class having same signature.
Behaviour	To Add/Extend more to method's behaviour.	To Change existing behaviour of method.
Polymorphism	Compile Time	Run Time
Inheritance	Not Required	Always Required
Method Signature	Must have different signature	Must have same signature.
Method Relationship	Relationship is between methods of same class.	Relationship is between methods of super class and sub class.
No of Classes	Does not require more than one class for overloading.	Requires at least 2 classes for overriding.

Any 6 points

Each point 1 Marks

6