CLASS XII
GUESS PAPER
COMPUTER SCIENCE

1).What is difference between in structure and class?

There are many differences between structure and class.

	structure	**class**
1	To creating structure we use **struct** keywords. (Structure is a collection of the different data type).there is no access type.	To creating class we use **class** keywords. (Class is a user defined data type which consists of attributes (data members) and behavior (member functions).By default the members of a class is private. It has following access type 1). Private 2) public 3) protected.
2	By the default structure nature is public type .it meaning we can access all data members and function of structure outside of structure and main function.	Class members are private in default .it meaning we cannot access all data members and function out side of class.
3	A structure contains only data.	While class bind both data and member functions.
4	Structure we can't initlisied the value to the variable.	but in class variable we assign the values
5	Structure doesn't support the polymorphism, inheritance and initialization. Because it not a parts of oops.	Class is oops based it supports data hiding, polymorphism, inheritance and initialization.
6	Struct student { Int roll_no; Char name[20]; }s1;	Class student { Private: Int roll; Char name[20]; Public: Void input();

		Void output(); };

2. What is difference between class and object?

A class is a type, and an object of this class is just a variable .But There are many difference between them.

	Class	Object
1	Class is a user defined data type with data members and member functions which can be public or privately accessed depending on access specifies.	**Object is an instance of a class.**
2	Class has only the logical existence.	While the object has a physical existence.
3	An Object on the other hand has a limited lifespan	Objects are created and eventually destroyed. Also during that lifetime, the attributes of the object may undergo significant change.
4	Object is an instance of a class. While a class can have many objects. Class is a static entity	While object is a dynamic entity.
5	Class(s) defines object	Object can't define classes
6	A Class is a blueprint of an object. class we declare.	object we call.

3. What is friend function? Explain with it example.

Ans :-

Its the main feature of a class that private member data of a class can be accessed only by the class' member functions. But there is an exception , A class can allow non-member functions and other classes to access its own private data, by making them as friends.

```
class Test
{
private:
   int a;
```

```
public:
   void xyz( );
   friend void display(Test);  //Friend of the class 'Test' , display() can access the
                               //private data members of the class
}

void display(Test x)
{
cout << x.a;// Class private data can be accessed.
}
```

4. What is a scope resolution operator? explain

A scope resolution operator (**::**), can be used to define the member functions of a class outside the class.

5. What do you mean by inheritance?

1).Inheritance is the property by which one class can acquire the properties of objects of another class.

2).Inheritance is the process of creating a new class called derived class from the existing class called base class.

3).Inheritance is the process of creating new classes, called derived classes, from existing classes or base classes. The derived class inherits all the capabilities of the base class, but can add embellishments and refinements of its own.

6. What is difference between polymorphism. Explain with example.

"Poly" means "many" and "morph" means "form". Polymorphism is the ability of an object (or reference) to assume (be replaced by) or become many different forms of object.

Example: function overloading, function overriding, virtual functions. Another example can be a plus '+' sign, used for adding two integers or for using it to concatenate two strings.

There are two types of polymorphisms. They are

1) Compile time Polymorphism

2) Runtime Polymorphism

In compile time polymorphism we have Function Overloading and Operator Overloading.

In Runtime polymorphism we have Virtual Functions and Dynamic Binding.

7. What is encapsulation?

1) Packaging an object's variables within its methods are called encapsulation.

2).The wrapping up of data and member function into a single unit is called encapsulation.

3).The data is not accessible to the outside world and only those functions which are wrapped into a class can access it.
4). Encapsulation is the concept to binding of data and functions. The data members are allowed to access by appropriate class functions or methods. Data members can't access from outside class.

8. What is abstraction?

1).Abstraction is of the process of hiding unwanted details from the user.

2).Abstraction is separating the logical properties from implementation details. For example driving the car is a logical property and design of the engine is the implementation detail.

3).Data Abstraction is process of separating the logical properties from its implementation.

(ADT)

9. What do you mean by inline function?

1).The inline functions are functions in a class with their function body.

2).When a function define the inside the body of the class is called Inline function or we can use just like general function only we add inline keyword in the function.

3). Inline function r used to tell the compiler to insert the code where the function call is made .It is upto the Compiler to decide whether to insert the code or not depending upon the code size and compiler optimized setting

10)what is difference between constructor and destructor

	constructor	destructor
1	Constructor is the member function of the class which has the same name as that of class and it is invoked whenever the class object is instantiated.	destructor is also the member function of same class name and has ~ operator when ever declared in the function and it is used to destruct the object which has been constructed ,whenever we want to destroy it
2	Constructor we can allocate memory.	Destructor is used for deallocate(release) the memory.
3	Constructor have three type in c++	There is no type of destructor in c++
4	It call automatic first in program, when object is created.	It call automatic in program at last.

Difference between "assignment operator" and a "copy constructor"

Copy constructor is called every time a copy of an object is made. When you pass an object by value, either into a function or as a function's return value, a temporary copy of that object is made.

Assignment operator is called whenever you assign to an object. Assignment operator must check to see if the right-hand side of the assignment operator is the object itself. It executes only the two sides are not equal.

Example.

```
Class A{/*...*/};

main()
{
  A a;
  A b = a; /* copy constructor will be called */
  A c;
  c = a; /* assignment operator will be called*/
}
```

Pass By Value example:

The function receives a copy of the variable. This local copy has scope, that is, exists only within the function. Any changes to the variable made in the function are not passed back to the calling routine. The advantages of passing by values are simplicity and that is guaranteed that the variable in the calling routine will be unchanged after return from the function. There are two main disadvantages. First, it is inefficient to make a copy of a variable, particularly if it is large such as an array, structure or class. Second, since the variable in the calling routine will not be modified even if that's what is desired, only way to pass information back to the calling routine is via the return value of the function. Only one value may be passed back this way.

Consider function void foo(int i);

int j=3;

foo(j);

Pass by Reference example:

C++ provides this third way to pass variables to a function. A reference in C++ is an alias to a variable. Any changes made to the reference will also be made to the original variable. When variables are passed into a function by reference, the modifications made by the function will be seen in the calling routine. References in C++ allow passing by reference like pointers do, but without the complicated notation. Since no local copies of the variables are made in the function, this technique is efficient. Additionally, passing multiple references into the function can modify

multiple variables.
Consider function void foo(int& i);

int j=3;

foo(&j);

Pass by Pointer example:

A pointer to the variable is passed to the function. The pointer can then be manipulated to change the value of the variable in the calling routine. The function cannot change the pointer itself since it gets a local copy of the pointer. However, the function can change the contents of memory, the variable, to which the pointer refers. The advantages of passing by pointer are that any changes to variables will be passed back to the calling routine and that multiple variables can be changed.

Consider function void foo(int* i);

int j=3;

int* p=&j;

foo(p);

if you look at pass by ref & pass by pointer it is almost the same but in pass by pointer you can do pointer arithmetic operation whereas in ref you cannot.

11).In c++ have a default constructor ?

1). Yes C++ does have a default constructor provided by the compiler.In this case all the members of the class are initialized to null values.These values act as the default values.For eg: MyClass me;In the above case since the object is not initialized to any value so the default constructor will be called which will initialize the class with the default values.

2). Yes C++ does have a default constructor provided by the compiler.In this case all the members of the class are initialized to null values.These values act as the default values.For eg: MyClass me;In the above case since the object is not initialized to any value so the default constructor will be called which will initialize the class with the default values.

3). yes, In C++ we have a default constructor.

A "default constructor" is a constructor that *can be called* with no arguments. One example of this is a constructor that takes no parameters

```
class Fred {
 public:
   Fred();   // Default constructor: can be called with no args
   ...
 };
```

12). why array index starts from 0[zero] only?

1) This boils down to the concept of Binary digits. Take an array size of 64 for example. We start from 0 and end at 63. We require 6 bits.But, if we were to start from 1 and end at 64, we would require 7 bits to store the same number, thus increasing the storage size.

2). because the name of an array is actually a pointer to the first element of this array in memory :array[0] is equivalent to *arrayarray[1] is equivalent to *(array + 1)...array[i] is equivalent to *(array + i)

3). C++ compilers are build such that. so array index always starts with 0.

13). what is memory leaking in c++ ?

1). Memory leak is - dynamically allocating memory and forgeting to free it. In C++, using a new operator to allocate a chunk of memory, and forgetting to delete it. There are several reasons this could occur. Some of them are,
1. Allocate a chunk of memory and assign the starting address to a pointer variable, and later, reassing another address to the same pointer without freeing its original address
2. Allocate memory for an array and not using proper delete method (delete[]) to free the memory
3. Unhandled exceptions - allocating memory and deleting it at the end of the program, but the program abnormally terminates (crashes) before the delete code line is executed.
2). Memory leak actually depends on the nature of the program. memory leaking means u store some data in memory but u forget the address of the block E.g

int *i = new int[2000]

delete i

u delete the address of the memory not hole memory

block

14). What is the difference between macro and inline()?

1. Inline follows strict parameter type checking, macros do not.

2. Macros are always expanded by preprocessor, whereas compiler may or may not replace the inline definitions.

3. Inline functions are similar to macros because they both are expanded at compile time, but the macros are expanded by the preprocessor, while inline functions are parsed by the compiler. There are several important differences: · Inline functions follow all the protocols of type safety enforced on normal functions. · Inline functions are specified using the same syntax as any other function except that they include the inline keyword in the function declaration. · Expressions passed as arguments to inline functions are evaluated once. In some cases, expressions passed as arguments to macros can be evaluated more than once.

15). What is difference between # defind amd macro?

#define

The #define directive takes two forms: defining a constant and creating a macro.

Defining a constant

#define *token* [*value*]

When defining a constant, you may optionally elect not to provide a value for that constant. In this case, the token will be replaced with blank text, but will be "defined" for the purposes of #ifdef and ifndef. If a value is provided, the given token will be replaced literally with the remainder of the text on the line. You should be careful when using #define in this way; see this article on the c preprocessor for a list of gotchas.

Defining a parameterized macro

```
#define token( [, s ... ]) statement
```

For instance, here is a standard way of writing a macro to return the max of two values.

```
#define MAX(a, b) ((a) > (b) ? (a) : (b))
```

Note that when writing macros there are a lot of small gotchas; you can read more about it here: the c preprocessor . To define a multiline macro, each line before the last should end with a \, which will result in a line continuation.

16). What is different between memberfunction and function? Explain.

When function is inside in class so it called memberfunction otherwise it called function.

17). What is type def and sizeof operator in c++?

sizeof Operator

The **sizeof** operator yields the size of its operand with respect to the size of type **char**.

Grammar

unary-expression:

sizeof *unary-expression*

sizeof (*type-name* **)**

The result of the **sizeof** operator is of type **size_t**, an integral type defined in the include file STDDEF.H. This operator allows you to avoid specifying machine-dependent data sizes in your programs.

The operand to **sizeof** can be one of the following:

- A type name. To use **sizeof** with a type name, the name must be enclosed in parentheses.
- An expression. When used with an expression, **sizeof** can be specified with or without the parentheses. The expression is not evaluated.

When the **sizeof** operator is applied to an object of type **char**, it yields 1. When the **sizeof** operator is applied to an array, it yields the total number of bytes in that array, not the size of the pointer represented by the array identifier. To obtain the size of the pointer represented by the array identifier, pass it as a parameter to a function that uses **sizeof**

typedef Specifier

A **typedef** declaration introduces a name that, within its scope, becomes a synonym for the type given by the *type-declaration* portion of the declaration.

Typedef int n;

18). What is Meta class? explain.

A class is consist of data member, data function and constructor and destructor and have mode private , public and protected , these combination of so many thing are called Meta class.

19). What is difference betweem multi level inheritance and multiple inheritance? explain

Multiple Inheritence : while a class has inheritted more than one classes then it is called multiple inheritance.

Multi-Level Inheritence : where a class can inherit only one class. while a class has inherited a class and it is being inherited by other class, this hierarchy is being called as Multi-Level Inheritence.

20). What is relationship in classes ? how many type relatioship in c++?

Relationship between classes is called inheritance. There are so many types ……

21). What is difference between data memberfunction and data members in c++?

Variable in class is called datamember and function in class is called datafunction.

22). Difference between base class and subclass in c++?explain

Which class is inherit is called base class and those inherit is called derived (subclass).

23). What is Actual parameter and formal parameter in c++?

Formal parameters are written in the function prototype and function header of the definition. Formal parameters are local variables which are assigned values from the arguments when the function is called.

When a function is *called*, the values (expressions) that are passed in the call are called the *arguments* or *actual parameters* (both terms mean the same thing). At the time of the call each actual parameter is assigned to the corresponding formal parameter in the function definition.

24). What is difference between Local & Global? explain

Local variable can access a particular area(limited areas). And global can access any where.

global call ::(scope revelation operator), but local call directly.

Creation of global is out side of function but local create always in side of function.

25). What is static data members and static functions in c++?

Classes can contain static member data and member functions. When a data member is declared as **static**, only one copy of the data is maintained for all objects of the class.

26).What is the Difference between Macro and ordinary definition?

1. Macro takes parameters where as ordinary definition does not.
2. Based on the parameter values to macro it can result in different value at run time. Ordinary defination value remains same at all place at run time.
3. Macro can be used for conditional operations where as definition can not.
4. Using macro one can achieve inline functionality in C ie. macro can be a function performing simple operations. This is not possible using definitions.