**1.** **What are the benefits of object-oriented programming over procedure-oriented programming? Describe the features of object-oriented programming. What is the task of const keyword?** *[4+4+2]*

**=>** The benefits of object-oriented programming over procedure-oriented programming are as follows;

* + Focus is given to data **rather** than procedure.
  + Programs are divided into objects which may communicate through functions.
  + Functions and data are tied together in a single unit.
  + Data can be hidden to prevent accidental modifications.
  + It also supports reusability of code along with inheritance, polymorphism, data abstraction and encapsulation.

The features of object-oriented programming are as follows;

* + Class  
    Classes are templates, model, design or blueprint that specifies data and operations. Once a class is defined, we can create any number of objects of its type. Each object is associated with the data of type class with which they are created.

For e.g.  


* + Object  
    Objects are defined as the physical instant of class. They can have attributes (properties) and method (behaviour). When an object is created, space for that object is allocated in primary memory.

For e.g.  


Here, Prafulla is an object of type Student (class).

* + Abstraction   
    Abstraction feature of OOP hides the internal details of how any object does its work. It only provides the interface to use the service that the object provides. A car is a great example of abstraction. You can start a car by turning the key or pressing the start button. You don't need to know how the engine is getting started, what all components your car has. The car internal implementation and complex logic is completely hidden from the user.
  + Inheritance  
    Inheritance is a process of creating new classes based on the existing classes. The new class acquires features from the existing class and adds more feature in it. The existing class is called base class and the newly created class is called derived class. The derived class inherits all the features except the private members of the base class. Thus, inheritance supports program reusability.

The task of const keyword is to declare or initialize constant variable or function i.e., whose value does not change throughout the program. E.g.   


Here, the value is a is constant i.e., a=30 throughout the program.

**2. List the features of C++. What are constructors, write their use and explain using an example.** *[4+6]*

**=>** The features of C++ are as follows;

* Classes
* Objects
* Abstraction
* Encapsulation
* Data hiding
* Inheritance
* Polymorphism
* Machine independent

Constructors are special member function of a class having same name as its class name with no return types which is invoked or accessed automatically when an object of the class containing the constructor is created. The main use of a constructor is to initialize some value to the data member of an object.

For e.g.

The output of this code is 

In the above program, Data( ) from line 8 is a constructor having same name as the class. It is called or invoked when the object is created i.e., in line 21 object(d) is created of type class (Data). This constructor initializes the value of its data members (a and b) to zero in order to avoid a garbage value. Only after this initialization, the increment value is safe from being affected by random garbage value. To sum up, a constructor is basically a function of a class having same name as the class and which is invoked each time an object of the class is created, generally to initialize a valid value.

**3. What is dynamic memory allocation? Write a C++ program to join two strings using dynamic constructor concept.** *[3+7]*

**=>** Dynamic memory allocation is the process of assigning or allocating memory space during the execution time or the run time with the aim of minimizing or properly utilizing the memory space.

Syntax:



**4. What is the disadvantage of using operator overloading in C++? Write a program to define a Class Distance with necessary data members and functions. Then overload the relational operators to compare the two objects of Distance class.** *[2+8]*

**=>** The disadvantage of using operator overloading in C++ are as follows;

* Precedence and Associativity of an operator cannot be changed.
* Basic meaning of operator can't be changed. That is, we cannot redefine the plus (+) operator to divide or subtract.
* Only existing operators can be overloaded. New operators cannot be created.
* The operator function for overloaded operator can't have default arguments.

**5. What is protected access specifier? Write a program with three classes students, test and results by using multilevel inheritance. Assume necessary data members and functions yourself and program with input information, input data and calculate marks total and display result.** *[3+7]*

**=>** The access specifier whose members cannot be directly accessed from outside of the class but can be inherited by a derived class is called protected access specifier. It is defined by the presence of ‘protected:’ keyword and is used in order to protect any data while allowing it to be inherited.



**6. List the features that are used in formatting the output. Explain each with example.**  *[10]*

**=>** The features that are used in formatting output are mentioned below;

1. width()  
   This function of ios class is used to define the width of the field to be used while displaying the output. It is normally accessed with cout object. E.g.;  
   Code:

1. precision()  
   This function belonging to ios class is used to specify maximum number of digits to be displayed as a whole in floating point number or the maximum number of digits to be displayed in the fractional part of the floating-point number. E.g.;  
   Code:  
   cout.precision(**3**);   
   cout<<sqrt(**2**)<<endl;   
   cout<<**1.2345678**;  
   Output:  
   1.412  
   1.234
2. fill()  
   This ios function is used to specify the character to be displayed in the unused portion of the display width. E.g.;  
   Code:  
   cout.fill(‘\*’);   
   cout.width(**7**);   
   cout<<**525**;   
   cout.fill();   
   Output:  
   \*\*\*\*525
3. setf()  
   Apart from above mentioned formatting, C++ also provides other ways of output formatting like left justified, scientific form, show positive sign, show base of displayed number etc. The ios member function setf() is used to set flags and bit-fields that controls the output in other ways as mentioned earlier.

Syntax:  
cout.setf(arg1,arg2);  
arg1->flag value  
arg2->bit field value  
For E.g.;  
Code:  
**int** x=-**456**;  
cout.setf(ios::internal,ios::adjustified);  
cout.width(**6**);  
cout.fill('#');  
cout<<x<<endl;  
Output:  
-##456

1. Unsetf()  
   This ios function is used to unset the specified flag value. For e.g.;  
   cout.unsetf(ios::showpos);  
   It clears the format bit corresponding to showpos. After this statement the program will not display ‘+’ sign in displaying positive number.

**7. Why do we need virtual function? Explain with suitable example. What is pure virtual function? What is the task of reinterpret cast operator?** *[6+2+2]*

**=>** Code as an example:

Here, pointer to base class is created and is assigned with the address of the base class first as shown in line 26. So, we get the first output “From Base Class Animal” when the function display( ) is called. Then we assign the address of other objects of derived class (Cow and Dog) but the output continues to remain same. This is because during compilation, pointer always selects the base class function as it is a pointer to base class object. The compiler is unknown about the address which is known at the run time and thus chooses the member function that matches the type of pointer due to which run time polymorphism cannot be achieved. Thus, we create a virtual function by using the keyword “virtual” in front of function name in base class so that desired output can be achieved and we can get runtime polymorphism.

The virtual function that has no body is called pure virtual function.

Syntax:

**class** **class\_name**{  
 **virtual** return\_type function\_name()=**0**;  
};

The static cast does not work with pointer. So, when we need to convert from one type to fundamentally different type i.e., from one pointer to other pointer or any other data type then reinterpret cat operator is used. To sum up, reinterpret cast operator handles conversions between unrelated types. It changes a pointer to any other type of pointer or data type and vice versa.

Syntax:

**reinterpret\_cast**<target\_type>(expression);

**8. Explain the importance of function template with suitable example. How default arguments can be used in class template? What are the tasks of try, catch and throw block?** *[4+3+3]*

**=>** 

In the above program we can find multiple swamp() functions which are used to swamp the two variables. Though this code swamps the variables but the code becomes redundant (as every function differs in data type of arguments) which can be eliminated by writing a single generic function that takes data as arguments and that generic function is template.



Like default arguments with normal functions the class template can also have a default argument associated with template parameter. The default template class argument is specified in the template declaration in the following form  
Syntax:  
  **template** <**class** **template\_type**=default\_data\_type>   
 class **class\_name**{  
 //data and functions declaration   
 };  
E.g.;  
 **template** <**class** **T**=**int**>   
 class **Test**{   
 //....   
 };

The primary purpose of function-try-blocks is to respond to an exception thrown from the member initializer list in a constructor by logging and rethrowing, modifying the exception object and rethrowing, throwing a different exception instead, or terminating the program.

The catch(…) form of the handler is used to catch all thrown exceptions that have not been caught by a previous catch block. The ellipsis in the catch argument indicates that any exception thrown can be handled by this handler.

The throw keyword throws an exception when a problem is detected, which lets us create a custom error. The catch statement allows you to define a block of code to be executed, if an error occurs in the try block.