

Q: 42: Write the names of two types of Templates. MARKS: 2

Answer:- (Page 256)

- a. Function Templates (in case we want to write general function like printArray)
- b. Class Templates (in case we want to write general class like Array class)

Q: 42: Describe the way to declare a template function as a friend of any class. 2marks

Answer:- (Page 294)

```
template< class T >
class B {
int data;
friend void doSomething( T ); // granting friendship to template
doSomething in // class B
friend A< T >; // granting friendship to class A in class B
};
```

Q: 43: Can a constructor throw an exception? How to handle the errors, when the constructor fails?

Answer:-

Yes, Constructor can throw an exception But In constructor if an exception is thrown than all objects created so far are destroyed. Exception due to constructor of any contained object or the constructor of a parent class can be caught in the member initialization list.

Q: 44: Why these operators can not be overloaded:

. , .* , ?: , ::

Answer:- (Page 141)

Reason: They take actual current object name, rather than value in their argument

Q: 46: What is graceful termination method of Error handling; give its example using C++.

Answer:- (Page 330)

Program can be designed in such a way that instead of abnormal termination, that causes the wastage of resources, program performs clean up tasks, mean we add check for expected errors (using if conditions),

Example – Graceful Termination

```
int Quotient (int a, int b ) {  
if(b == 0){  
cout << "Denominator can't " << " be zero" << endl;  
// Do local clean up  
exit(1);  
}  
return a / b;  
}
```

Q: 47: How can you describe Virtual destructor? Why Virtual destructor is used?

Answer:- (Page 234)

When delete operator is applied to a base class pointer, base class destructor is called regardless of the object type. Only base class part of object will be deleted other parts will remain as it is this result in memory leak (wastage of memory), Virtual Destructors make it possible that complete object is being deleted so there is no memory leak (waste of memory).

Q: 48: Motor car can be called a good candidate for being an object so, this object contains Encapsulation characteristic. You are required to apply the concept of Encapsulation on "Motor Car".

Answer:-

Yes Motor Car is a good candidate for apply the concept of encapsulation

In Motor Car we have the following Objects which fully describes the phenomena of encapsulation.

1. Steer Wheels
2. Accelerate
3. Change Gear
4. Apply Brakes
5. Turn Lights On/Off

In encapsulation we just provide the user an interface and hide the Full implementation from the user because user don't want to know the implementation he Just want his work should be done perfectly

For example: Breaks of car, we gave just a paddle to the User and encapsulate how it works inside the engine. Same for accelerator, gear, steer wheels and turn on and off of Light.

Q: 49: Suppose base class and derived class has a member function with same signature. Now call that member function through the pointers to a base class object. Now, what determines that the base class member function or derived class function will be called?

Q: 51: What do you know about exception in initialization? Explain with example.

Answer:- (Page 345)

Exception in Initialization List

Exception due to constructor of any contained object or the constructor of a parent class can be caught in the member initialization list.

Example

```
Student::Student (String aName) : name(aName)
/*The constructor of String can throw a exception*/
{
...
}
```

Exception in Initialization List

The programmer may want to catch the exception and perform some action to rectify the problem

Example

```
Student::Student (String aName)
try : name(aName) {
...
}
catch (...) {
}
```

1- Resolve problems in overloading assignment operator in string class.

Answer:-

The problem in statement

`str1 = str2 = str3` is,

`str1=str2=str3` is resolved as:

`str1.operator=(str2.operator=(str3))`

Assignment operator is being called two times one for part `str2 = str3` and then for `str1 = (str2 = str3)` as assignment operator is right associate so first `str2=str3` will be executed, and `str2` will become equal to `str3`, then first overloaded assignment operator execution result will be assigned to `s1`,

`str1.operator=(str2.operator=(str3))`

2- What is random-iterator? What relation b/w random iterator and vector 5 marks

Answer:-

Random Access Iterators:-

They have all the capabilities of bidirectional Iterators plus they can directly access any element of a container.

You can think of vectors as smart arrays. They manage storage allocation for you, expanding and contracting the size of the vector as you insert or erase data. You can use vectors much like arrays, accessing elements with the `[]` operator. Such random access is very fast with vectors. as we can access any element of vector using its index so we can use random access iterator.

It's also fast to add (or push) a new data item onto the end (the back) of the vector. When this happens, the vector's size is automatically increased to hold the new item.

3- Class complex {double real,ing public }; overload the *= operator for the complex class by writing C++ code 3 marks

```
class Shape {  
...  
public:  
virtual void draw() = 0;  
}
```

4. Write down a list of four intangible objects?

Answer:-

1. Time
2. Light
3. intelligence
4. Temperature
5. software

5. Give the pseudo code of non case sensitive comparison function of string class.

Answer:- (Page 263)

```
int caseSencompare( char* str1, char* str2 )  
{  
for (int i = 0; i < strlen( str1 ) && i < strlen( str2 ); ++i)  
if ( str1[i] != str2[i] )  
return str1[i] - str2[i];  
return strlen(str1) - strlen(str2);  
}
```

6. Describe three properties necessary a container to implement generic algorithms.

Answer:- (Page 301)

We claimed that this algorithm is generic, because it works for any aggregate object (container) that defines following three operations

- a. Increment operator (++)
- b. Dereferencing operator (*)
- c. Inequality operator (!=)

7. Mobile is a good entity for being an object. Write characteristics, behavior and unique ID.

Answer:-

Characteristics: Buttons, Battery, Screen

Behavior: Calling, Sending Message, Internet, Multimedia

Unique ID: Nokia

8. What are the container requirements? Write details.

Answer:-

As each container need to perform certain operations on the elements added to it like their copy while creating another instance of container or their comparison while performing certain operation on them like their sorting , so the elements that we are going to add in containers should provide this kind of basic functionality.

Examples of these functionalities are given below,

Answer:- (Page 153)

```
class Complex{  
double real, img;  
public:  
Complex & operator+=(const Complex & rhs);  
Complex & operator+=(count double & rhs);  
...  
};
```

4-suppose there is a class Doctor with data member. Name Department and Doctor. you are required to write default copy constructor as well as deep copy constructor in order to initialize object and explain the difference with the help of main() function. Only write up to constructor code and main() function 5 marks