Date _____
Page _____

Ques-1 what is the difference between deep copy and shallow copy and how does C++ handle them by default.

Ans shallow copy

A shallow copy copies the values of data members from one object to another, including pointers.

In case of pointers only the addresses are copied not the actual data.

It means both objects will point to same memory location, which can cause problems too.

Deep copy

It creates a separate copy of dynamically allocated memory.

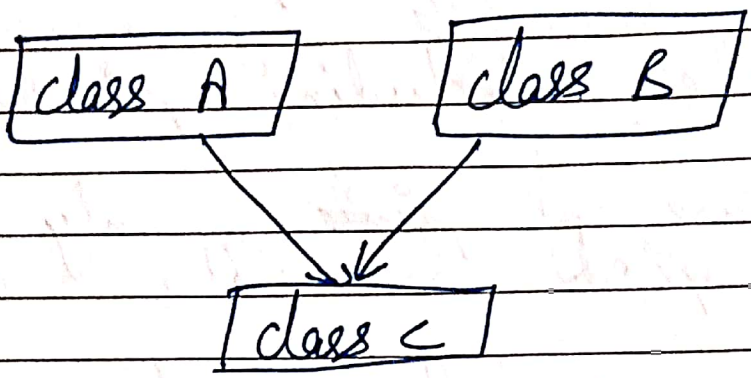
Both objects have their own copies of data, so deleting something will not affect the other.

By default C++ uses a shallow copy when copy constructor is not explicitly defined.

DATE
PAGE

Ques-3 How does multiple inheritance works in C++ ? What is diamond problem and how is it resolved ?

Ans. Multiple inheritance means a class can be inherit from more than one base class.



Here 'class A' and 'class B' are base or parent class and 'class C' is derived class.

* Diamond problem

occurs when a class inherits from two classes that share a common base or parent class.

when multiple inheritance forms a diamond shape hierarchy then ambiguity arises.

Solution

To avoid duplicate copies of the base class, C++ provides virtual inheritance, so that only one shared copy of the common base class exists, removing ambiguity.

Ques-3 Difference between virtual and pure virtual functions.

Virtual functions

It is a function in a base class that can be overridden in a derived class.

It supports run time polymorphism.

If the derived class doesn't override it, the base class version will be used.

Pure virtual function

It is a virtual function with no implementation in the base class.

Declared by assigning =0 in the base class.

A class containing at least one pure virtual function becomes an abstract class.

DATE
PAGE

Ques-4 what are dangling pointers and how they can be avoided.

A dangling pointer is a pointer that still points to a memory location which has been free or deleted.

Solutions

→ set pointer to nullptr after deleting

→ avoid returning addresses of local variable.

Ques-5 Difference between malloc/free and new/delete.

* malloc/free

→ It comes from 'c' language.

→ used only for raw memory allocation/deallocation.

→ Do not call constructors or destructors.

DATE _____
PAGE _____

new / delete

→ It is introduced in c++.

→ allocate memory and call the constructor.

→ delete free the memory and calls the destructor.

Ques-6 Difference between map and multimap.

Ans It is a container that stores key value pairs.

each key will be unique. It cannot be duplicated.

Syntax

```
map<int, string> mp;
```

Multimap

It is similar to map, but allows multiple elements with the same key.

Keys are not unique.

Date
Page

Syntax

multimap <int, string> mmp;.

Ques 7 Difference between std: move and std: forward.

Ans std: move.

It says "I dont need this anymore, you can move from it.

After moving, the old object is still valid but usually empty.

std: forward

→ used mostly in template functions.

→ It keeps the original form of what you pass. If it was an lvalue, stays lvalue, if it was an rvalue, stays rvalue.

→ Helpful in writing generic wrapper functions.

Ques-8 Difference between unique_ptr, shared_ptr, and weak_ptr.

Ans ① unique_ptr

→ In this only one owner is present.
→ Copy is not allowed.
→ It can only moved.
→ When the unique_ptr goes out of scope → it deletes the object automatically.

② shared_ptr

→ In this multiple owner are present
→ copy is allowed here.
→ It keeps reference count.

③ weak_ptr

→ No ownership
→ copy is allowed
→ Does not affect reference count.
→ Does not control lifetime.