Deep Copy vs Shallow Copy Constructors

Deep Copy:

- deep copy constructor is useful for making a copy of dynamic structures which are in heap. like Dynamic Array ,Linked List, Trees etc.
- if member of a class is a pointer 'p' and it is pointing to array or object in heap.
- if object A is having its member 'p' pointing on an array or object in heap and you are making a copy of A as B then 'p' of B will also point on same array or object of A.
- to have separate copy of B another array or object should be created and make 'p' point on it.
- if a copy constructor is doing it, we say Deep copy constructor.

Shallow Copy:

- Copy constructor will copy only the members of an object.
- If an object is holding any memory in heap, it will not be copied.

Copy constructor vs Assignment operator

their working is same but usage is different.

Rectangle r1(10,5);

Rectangle r2(r1); // copy constructor is called

Rectangle r3=r1; // assignment operator is called

Create object in Stack or Heap

Stack:

Rectangle r1; // Valid

Rectangle r1( ); // invalid, don't give empty brackets.

Heap:

Rectangle *p; // pointer, it is created in stack.

p=new Rectangle(); // object is created in heap. Empty () can be given.

Deep Copy vs Shallow Copy Constructors

Deep Copy:

- deep copy constructor is useful for making a copy of dynamic structures which are in heap. like Dynamic Array ,Linked List, Trees etc.
- if member of a class is a pointer 'p' and it is pointing to array or object in heap.
- if object A is having its member 'p' pointing on an array or object in heap and you are making a copy of A as B then 'p' of B will also point on same array or object of A.
- to have separate copy of B another array or object should be created and make 'p' point on it.
- if a copy constructor is doing it, we say Deep copy constructor.

Shallow Copy:

- Copy constructor will copy only the members of an object.
- If an object is holding any memory in heap, it will not be copied.

Copy constructor vs Assignment operator

their working is same but usage is different.

Rectangle r1(10,5);

Rectangle r2(r1); // copy constructor is called

Rectangle r3=r1; // assignment operator is called

What is a default constructor?

- A constructor which dent take any parameters is called as default constructor.
- There are 2 Types of default constructors

1. Compiler provided
2. User-defined

Compiler provided default constructor

- If we don’t write any constructor, compiler will provide a default constructor.
- It will create the object, but doesn’t initialise data members.
- If we define our own contractor, then compiler will not provide default constructor.

Why we write constructor?

- We can initialise the data members of an object.
- If data members are initialised then we can use the object.

Type of Constructors

There are 3 type of constructors.

1. Non-parametrised (also called as default)
2. Parameterised
3. Copy constructor

Do we have to write all constructors ?

- It is better to write all constructors.

- I am not writing it in every lecture because I have to focus on actual topic.

Why copy constructor should take reference?

- If parameter is call by value then it will create new object for parameter.
- Constructor will call constructor again to create the object of parameter.
- It will be become a recursive call to constructor.

Can we initialise the variables directly?

- Yes you can initialise. But values will always be same.
- Constructor will allow us to initialise with desired values.

Can we read values inside the constructor using cin?

- If main() is creating object then, it is better to read values inside main and pass values to constructor.
- Using cin means, interacting with user. If user interaction is done in main() then it is a good design.