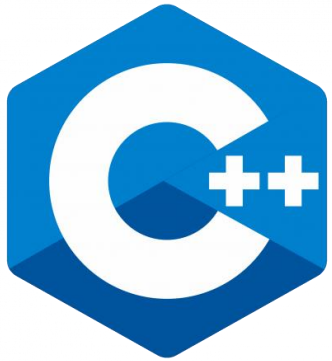


C++ Programming

Trainer : Pradnyaa S. Dindorkar

Email: pradnya@sunbeaminfo.com

We did…..

1. Modular Approach
2. Constant
3. References
4. Difference between Pointers and reference
5. Sum function and Copy Constructor

Todays Topics….

1. New and delete
2. Difference between New and malloc
3. Shallow Copy and deep copy
4. Static concept
5. Friend Function

Dynamic Memory Allocation

To allocate memory dynamically then we should use **new** operator

To deallocate that memory we should use **delete** operator.

Dangling pointer :- The pointer which contains, address of deallocated memory.

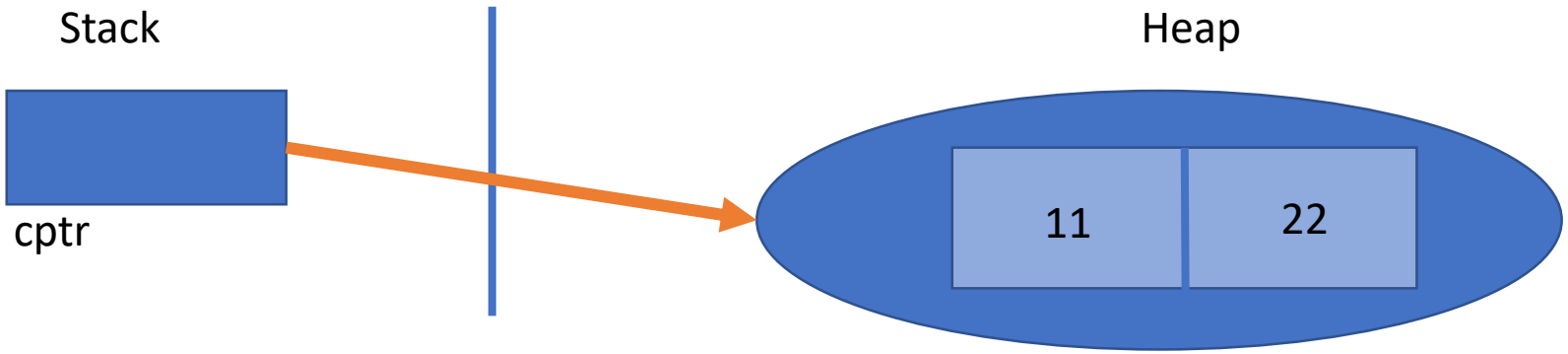
Memory leakage :- When we allocate space in memory, and if we loose pointer to reach to that memory then such wastage of memory is called memory leakage.

```
int main()
{
        int *ptr = new int;             // allocate memory
        *ptr = 125;                     //Dereferencing
        cout<<"Value:"<<*ptr<<endl;  //Dereferencing
        delete ptr;                     // deallocate memory
        ptr = NULL;
        return 0;
}
```

Heap Based object

```
Complex *cptr=new Complex (11,22) ;
delete cptr;
```

- By using new we are allocating dynamic memory for complex class object .
- Object get created on heap section hence this object is call Heap based or dynamic object.
- Cptr is complex type pointer which is holding the address of that dynamic object.



Difference between malloc and new

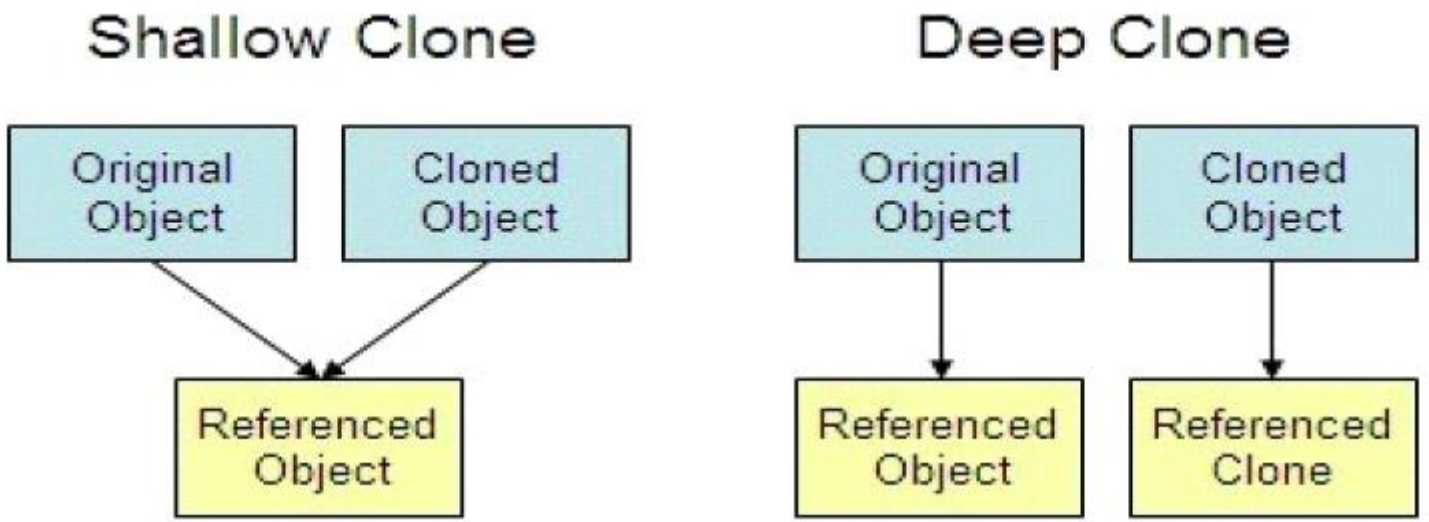
new	malloc
new is an operator.	malloc is a function.
new returns a typecasted Pointer, so no need to do explicit typecasting.	malloc returns void pointer, malloc returns void pointer, cast it explicitly, before use.
We must mention the datatype while allocating the memory with new.	malloc accepts only the exact no. of bytes required, so no need to mention datatype.
When memory is allocated with new, constructor gets called for the object.	When memory gets allocated by malloc function, constructor function does not gets called.

Difference between malloc and new

new	malloc
Memory allocated by new is released by the operator delete.	Memory allocated by malloc is released by function free.
Destructor is called when memory is released with delete.	Destructor is NOT called when the memory is released with free.
To release memory for array syntax is delete[ ]	To release memory for array syntax is free(ptr);
In case new fails to allocate memory, it raises a Run time exception called as bad_alloc.	In case malloc fails to allocate memory it returns a NULL.

Object Copying

- In object-oriented programming, “object copying” is a process of creating a copy of an existing object.
- The resulting object is called an object copy or simply copy of the original object.
- Methods of copying:
  - Shallow copy
  - Deep copy



Types of Copy

The copy constructor is used to initialize the new object with the previously created object of the same class.

- **Shallow Copy**
  - The process of copying state of object into another object.
  - It is also called as **bit-wise** copy.
  - When we assign one object to another object at that time copying all the contents from source object to destination object as it is. Such type of copy is called as shallow copy.
  - Compiler by default create a shallow copy. Default copy constructor always create shallow copy.

- **Deep Copy**
  - Deep copy is the process of copying state of the object by modifying some state.
  - It is also called as **member-wise** copy.
  - When class contains at least one data member of pointer type, and class contains user defined destructor then when we assign one object to another object then copy constructor is called.
  - At that time instead of copy base address allocate a new memory for newly created object and then copy contain from memory of source object into memory of destination object. Such type of copy is called as deep copy.

Static Variable

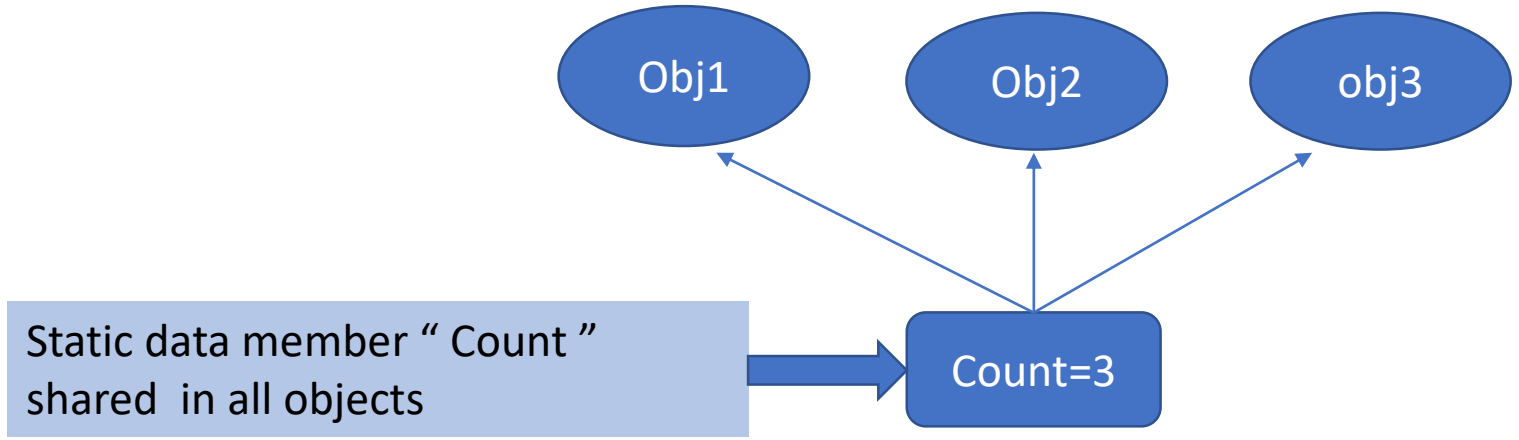
- All the static and global variables get space only once during program loading / before starting execution of main function
- Static variable is also called as shared variable.
- Initialized static and global variable get space on Data segment.
- Default value of static and global variable is zero.
- Static variables are same as global variables but it is having limited scope.

Static Data member

- If we want to share value of data member between all objects of same class then we should declare that data member as static data member.

- It is mandatory to provide global definition of static data member otherwise linker generates error.

- Static data member get space during class loading per class so it is called as **class-level variable**



Static Member Function

- Except main function, we can declare global function as well as member function static.
- static members of the class we should declare member function **static**.
- static member function is also called as **class level method.**
- To access class level method we should use classname and ::(scope resolution) operator.
- This pointer is not available in static member function .

Friend :-

- A non member function of a class which designed to access private data of a class is called friend function.

- To declare a function as a friend of a class, precede the function prototype in the class definition with keyword **friend**

```
class MyData
{
private:
        int pin;
        int pass;

public:
        MyData(int pin,int pass);
        void PrintMyAccDetails();
        friend void anyFunction();
};
```

```
void anyFunction()
{
        MyData d1;
        d1.pass=9898;
        d1.pin=9999;
        d1.PrintMyAccDetails();
}
```

C++ is not a pure Object Oriented Language Because

- You can write code without creating a class in C++, and main() is a global function.
- Support primitive data types, e.g., int, char, etc. Instances(variable) of these primitive types are NOT objects.
- C++ provides "Friend" which is absolute corruption to the OO-Principle of encapsulation.

Thank You