C++ Programming – Lecture 6

Constructor Nuances

- 0-argument constructor is also called default constructor, since compiler provides one if we not.
- Constructors are created based on the need. If we are not creating objects that need 0-argument constructor, then there is no need to provide such a constructor in the class.
- Constructors that take one or more arguments are called parameterized constructors.
- If an object can be built using multiple constructors then it creates a conflict as to which one should be used. Hence compiler flags an error.
- Initializer lists offer an alternate syntax for creation of constructors.

  ```
  Comp ( double  x, double  y ) : r ( x ), i ( y )
  {
  }
  ```

Static and Dynamic Memory Allocation

- In static memory allocation decision about how many bytes to allocate and where to allocate them is taken at compilation stage.
- In dynamic memory allocation decisions about how many bytes to allocate and where to allocate them is taken at execution stage.
- Stack and heap are regions in memory.
- Heap is also called free store.
- Static memory allocation allocates space in stack.
- Dynamic memory allocation allocates space in heap.
- It is a good practice to create normal variables in stack, as they take less memory space and they are created and eliminated quickly with no overheads.
- It is a good practice to create arrays, structures and objects in heap, as they take more memory space.

Named Object Creation and Deletion

- Named objects can be created statically on the stack. Ex. Complex c1, c2 ;
- On creation of named object constructor gets called.
- Named objects cannot be deleted dynamically.
- Named objects die when control goes out of the function in which they are created.

- If named object is global, then it dies when the execution of the program comes to an end.

Nameless Object Creation and Deletion

- Nameless objects can be created at runtime using new operator.

  Complex ptrc1 = new Complex ;

  Complex ptrc2 = new Complex (1.3, 2.1) ;
- new allocates memory for object on heap dynamically, calls constructor and returns appropriate type of pointer. Objects created in heap are nameless.
- Nameless objects can be deleted at runtime using delete operator. For example,
- delete ptrc1 ;
- delete ptrc2 ;
- delete calls destructor and deallocates memory for object on heap dynamically.
- delete ptrc1 deletes object pointed to by ptrc1. It does not delete ptrc1.

Memory Leak and Dangling Pointer

- Memory leak means when memory stands allocated but we have no way to use it or de-allocate it.
- If new is used in the constructor, use delete in the destructor. If we don't do this a memory leak is bound to occur when the object dies.
- After deleting the object it is a good practice to set ptr to NULL. If we don't do this ptr becomes a dangling pointer.

Manager Functions

- Compiler provides 4 manager functions:

  0-Arg Constructor - Empty
  Destructor - Empty
  Copy Constructor - Contains code to copy contents of one object into another
  Overloaded = operator function - Contains code to copy contents of one object into another
- If an object contains a pointer, we must use new in constructor to set up this pointer.
- If we use new in the constructor, we must define our own destructor to ensure proper deletion of object.