C++ Programming

Trainer : Rohan Paramane

Email: rohan.paramane@sunbeaminfo.com

Escape Sequence and Manipulators

- **Manipulators** are helping functions that can modify the input/output stream.
- It does not mean that we change the value of a variable, it only modifies the I/O stream using insertion (<<) and extraction (>>) operators.
- Header File : #include<iomanip> // input output manipulation
  - Setbase(16) or hex
  - setbase(8) or oct
  - setbase(10) or dec
  - Endl

- **Escape Sequences**
  - \b , \t , \n , \\, \' , \"

- **setw (val)**
- **setfill(char c)**
- **setprecision (val)**

- **Example :**

```
double f =3.14159;
cout<< std::setprecision(5) << f <<'\n';
Int Num= 16;
cout<<hex<<num;
```

Scope

- It decides area/region/boundry in which we can access the element.
- **Types of scope in C++:**
  1. Block scope
  2. Function scope
  3. Prototype scope
  4. Class scope
  5. Namespace scope
  6. File scope
  7. Program scope

Example Scope

```
int num6;               //Program Scope
static int num5;                //File Scope
```

```
namespace ntest {
    int num4;                   //Namespace scope

    class Test {
    int num3;                   //Class Scope
};
}
```

```
int main( void ) {
            int num1 = 10; //Function Scope
            while( true ) {
            int temp = 0;    //Block Scope
}
            return 0;
}
```

```
void sum( int num1, int num2 ); //Prototype scope
```

Macro

- Symbolic constant is called as macro
- Expanding macro is a job of pre processor.

- Example:
  - #define SIZE 10
  - #define EOF -1
  - #define MULTIPLY(x,y) x*y

- Few other Macro's in C++
  - __FILE__, __LINE__,__DATE__,__TIME__

Singleton Design Pattern

- Singleton class is the class whose only one object can be created.
- If we try to create second one, we get reference of the first object only.
- It is one of the design pattern
- Since constructor is private, object can be created only through static function

```
class Single {
private:
Single () { }
static Single *ptr;
public:
static Single* create()
{  if(ptr==NULL)
   ptr=new Single();
   return ptr;          }
}
Single* Single::ptr=0;
```

Local & Nested Class

- **Local Class**
  - If inside a function you declare a class then such classes are called as local classes.
  - Inside local class you can access static and global members but you cannot access the local members declared inside the function where the class is declared.

- **Nested class**
  - A class declared inside another class is called as nested class
  - A nested class can access all the private and public members of outer class directly on the outer class object
  - An outer class can access only public members of nested inner class on its object.

Thank You