C++ Notes – BCA – 3rd Semester

Static Keyword (Part – II)

Static Keyword –

- **Static Variables in Functions –**
  1. Static variables when used inside function are initialized only once, and then they hold there value even through function calls.
  2. These static variables are stored on static storage area , not in stack.

e.g.,

```
#include<iostream.h>
#include<conio.h>

void display()
{
    static int p = 99;
    cout<<“Value is: ”<<p;
}
```

```
void main()
{
    display();
}
```

Static Keyword –

- **Static Member Variable in Class –**
  1. Static data members of class are those members which are shared by all the objects.
  2. Static data member has a single piece of storage, and is not available as separate copy with each object, like other non-static data members.
  3. Static member variables (data members) are not initialized using constructor, because these are not dependent on object initialization.
  4. It must be initialized explicitly, always outside the class. If not initialized, Linker will give error.

```
#include<iostream.h>
#include<conio.h>

class X
{
        public:
        static int i;
        X()
       {

        }
};

int X :: i=1;
```

```
void main()
{
        X obj;
        cout << obj.i;
}
```

Static Keyword –

- **Static Methods in Class –**
  1. Just like the static data members or static variables inside the class, static member functions also does not depend on object of class.
  2. We are allowed to invoke a static member function using the object and the ‘.’ operator but it is recommended to invoke the static members using the class name and the scope resolution operator.
  3. Static member functions are allowed to access only the static data members or other static member functions, they can not access the non-static data members or member functions of the class.

```
#include<iostream.h>
#include<conio.h>

class Sample
{
        public:
                static void printMsg()
                {
                        cout<<"Welcome to GGI";
                }
};
```

```
void main()
{
        Sample :: printMsg();
}
```

Static Keyword –

- **Static Class Objects –**
  1. Static keyword works in the same way for class objects too.
  2. Objects declared static are allocated storage in static storage area, and have scope till the end of program.
  3. Static objects are also initialized using constructors like other normal objects. Assignment to zero, on using static keyword is only for primitive data types, not for user defined data type.

```
class Abc
{
        int i;
        public:
        Abc()
        {
                i=0;
                cout << "constructor";
        }
        ~Abc()
        {
                cout << "destructor";
        }
};
```

```
void f()
{
        static Abc obj;
}

void main()
{
        int x=0;
        if(x==0)
        {
                f();
        }
}
```

THANK YOU…!