**Static Variables**

Whatever values the function puts into its static local variables during one call will still be present when the function is called again.

**Example:**

void s\_func()

{

static int x=0;

x++;

cout<<“x=”<<x<<endl;

}

int main()

{

s\_func();

s\_func();

s\_func();

}

**OUTPUT:**

1

2

3

**STATIC DATA MEMBERS OF CLASSES**

Consider the following class

class Rectangle

{

Private:

int length;

int width;

public:

Rectangle(int =1,int=1);

void print();

};

Rectangle R1;

Rectangle R2(1,5);

The length and width of R1 is independent of the length and width of R2 and is stored in different memory location. Changes made to R1’s length do not affect R2’s length and vice versa.

If we want all the instances (objects) of class to share data, we can use static variables. Static variables store values for the variables in a common memory location. If one object changes the value of that variable, all other objects are also affected.

class Rectangle

{

Private:

int length;

int width;

static int Num\_of\_objects;

public:

Rectangle(int =1,int=1);

void printValues();

};

int Rectangle:: Num\_of\_objects=0; //initialization of static variable

Rectangle::Rectangle(int a, int b)

{

length=a; width=b;

Num\_of\_objects++;

}

void Rectangle::printValues()

{cout<<“length=“<<length<<endl;

cout<<“width=“<<width;

cout<<”Total Objects”<<Num\_of\_objects;

}

**Num\_of\_Objects** is incremented whenever an object is created.

Consider following main()

Output:

length=2 width=3

Total Objects 2

void main()

{

Rectangle R1(2,3);

Rectangle R2(3,4);

R1.printValues();

}

**R1**

Length

Width

2

3

**R1**

Length

Width

3

4

Num\_of\_Objects

**STATIC FUNCTION** *(Returns a static variable)*

class Rectangle

{

private:

int length;

int width;

static int Num\_of\_objects;

public:

Rectangle(int =1,int=1);

void printValues();

static int GetValue();

};

int Rectangle::Num\_of\_objects=0; //initialization of static variable

int Rectangle::GetValue()

{

return Num\_of\_objects;

}

Rectangle::Rectangle(int a, int b)

{

length=a;

width=b;

Num\_of\_objects++;

}

void Rectangle::printValues() {

cout<<"length= "<<length<<"\t";

cout<<"width= "<<width<<endl;

}

int main()

{

Rectangle R1(2,3);

Rectangle R2(5,4);

cout<<Rectangle::GetValue();

return 0;

}

**Output:**

2

**Character pointers as private data members**

#include <cstring>

class Employee

{

private:

Char \*Name;

int ID;

public:

Employee( char \*=NULL , int =0);

}

Employee::Employee(char \* N, int id)

{

ID=id;

int len=strlen(N)+1;

Name=new char[len];

strcpy(Name, N);

}

int main()

{

Employee E1(“Ahmad Khan”, 12);

}

**EXAMPLE: class MATRIX**

class matrix

{

int \*\*p;

int row, col;

public:

* **Constructor with default parameters**
* **Copy constructor**
* **Destructor**
* **Input function**
* **Output function**

};