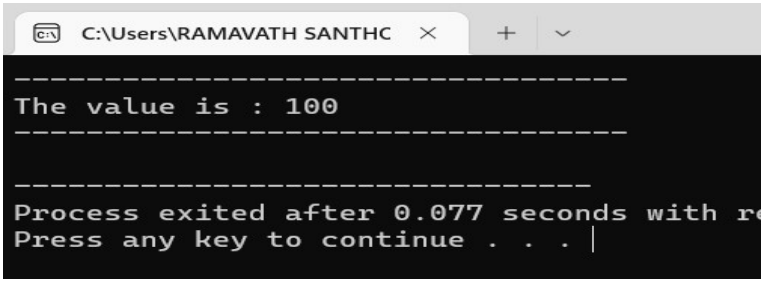
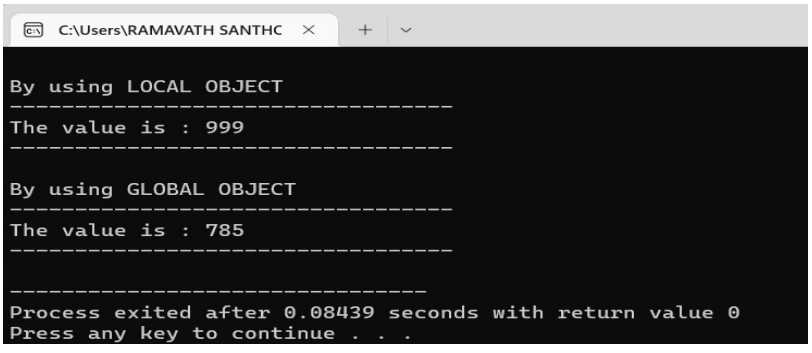
ASSIGNMENT 7
CONSTRUCTORS AND CLASS

```
/*
Q1  Write a program that uses a class where the member functions are defined outside a
class.
*/
#include <iostream>
using namespace std;
class Example
{

        int val;
        public:
        //function declarations
        void init_val(int v);
        void print_val();
};
//function definitions
void Example::init_val(int v)
{
   val=v;
}
void Example::print_val()
{
        cout<<"---------------------------------\n";
   cout<<"The value is : "<<val<<endl;
   cout<<"---------------------------------\n";
}
int main()
{
   //create object
        Example Ex;
        Ex.init_val(100);
        Ex.print_val();
        return 0;
}
```



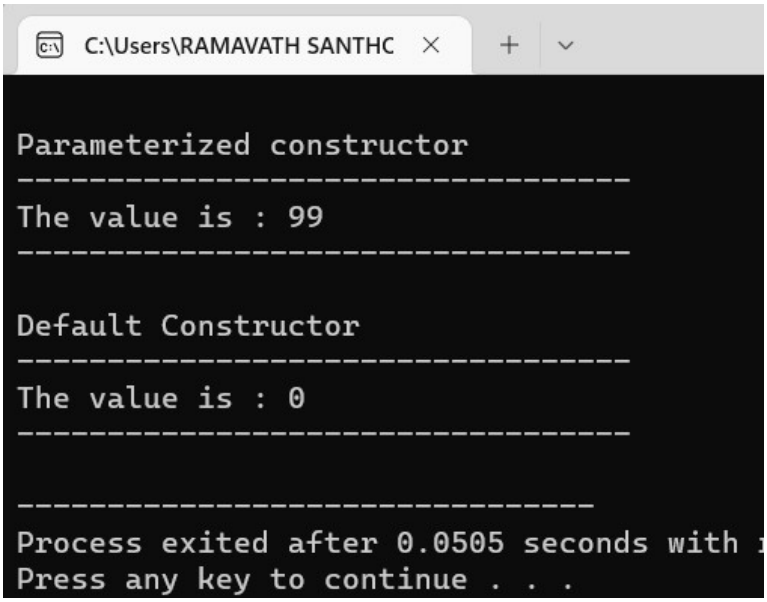

```
/* Q2  Try with local and global objects (with same program Q1)*/
#include <iostream>
```

```
using namespace std;
class Example
{
    private:
          int val;
          public:
          //function declarations
          void init_val(int v);
          void print_val();
};
    //function definitions

void Example::init_val(int v)
{
   val=v;
}
void Example::print_val()
{
    cout<<"---------------------------------\n";
    cout<<"The value is : "<<val<<endl;
    cout<<"---------------------------------\n";
}

Example obj; // global object
int main()
{
   //create object
   Example Ex; //local obj
   cout << "\nBy using LOCAL OBJECT"<<endl;
   Ex.init_val(999);
   Ex.print_val();
   cout << "\nBy using GLOBAL OBJECT"<<endl;
   obj.init_val(785);
   obj.print_val();
   return 0;
}
```



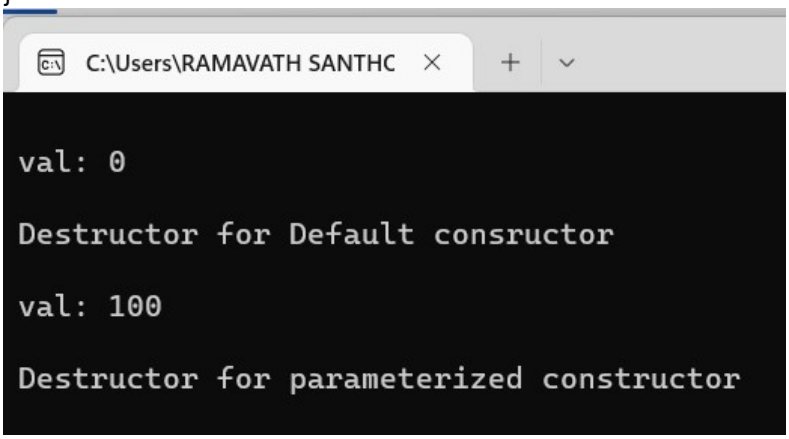

```
/*
Q3  Try with different constructors and definition inside &amp;amp; outside of the class (with
same
program Q1)
*/
```

```
#include <iostream>
using namespace std;
class Example
{
    private:
    int val;
            public:
            //constructor declarations
            Example()
            {
            val = 0;
            }
            Example(int);
            void print_val();
 };
    //constructor definitions
Example :: Example(int v)
{
   val=v;
}
void Example::print_val()
{
    cout<<"---------------------------------\n";
    cout<<"The value is : "<<val<<endl;
    cout<<"---------------------------------\n";
}

Example obj; // global object
int main()
{
   //create object
   cout << "\nParameterized constructor"<<endl;
   Example Ex(99); //local obj
   Ex.print_val();
   cout << "\nDefault Constructor"<<endl;
   Example obj;
   obj.print_val();
   return 0;
}
```



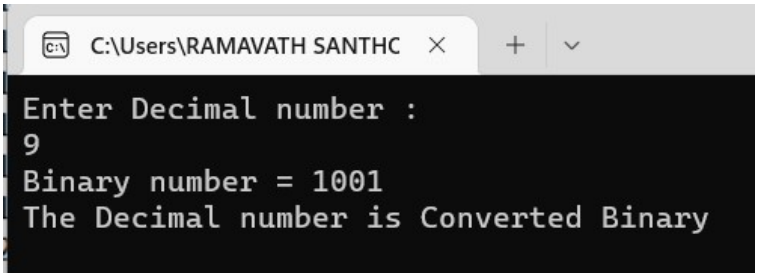

```
/*
Q4) Try with distractor (with same program Q1)
*/
#include <iostream>
using namespace std;
class Example
{
  private:
  int val;
  public:
//constructor declarations
  Example()
  {
  val = 0;
  }
  Example(int);
  ~Example();
};

//constructor definitions
Example :: Example(int v)
{
  val=v;
}
Example :: ~Example()
{
  cout<<"\nval: "<<val<<endl;
  if(val == 100)
  cout << "\nDestructor for parameterized constructor"<<endl;
  if(val == 0)
  cout << "\nDestructor for Default consructor"<<endl;
}
```

```
int main()
{
//create object
   Example Ex(100); //local obj
   Example obj;
   return 0;
}
```

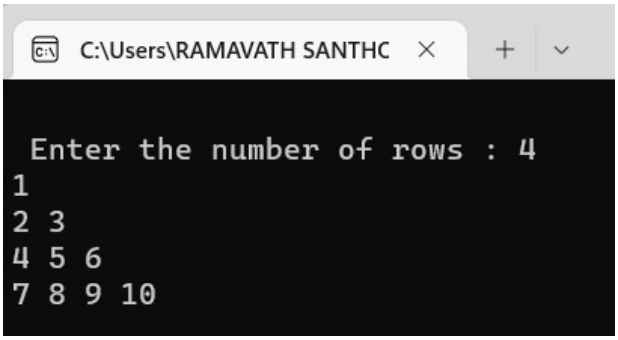



```
/*
Q5) Write a program in C++ to convert a decimal number into binary without using an
array and using the constructor and destructor.
*/
#include <iostream>
using namespace std;
class dec_to_bin
{
    public:
    dec_to_bin()
          {
          int p=1,bin=0,n;
          cout<<"Enter Decimal number : ";
          cin>>n;
          while(n)
          {
          bin += (n%2)*p;
          p=p*10;
          n=n/2;
          }
          cout<<"binary number = "<< bin;
          }
          ~dec_to_bin()
          {
          cout << "\nDecimal to Binary Converted";
          }
};
int main()
{
```

```
    dec_to_bin obj;
    return 0;
}
```

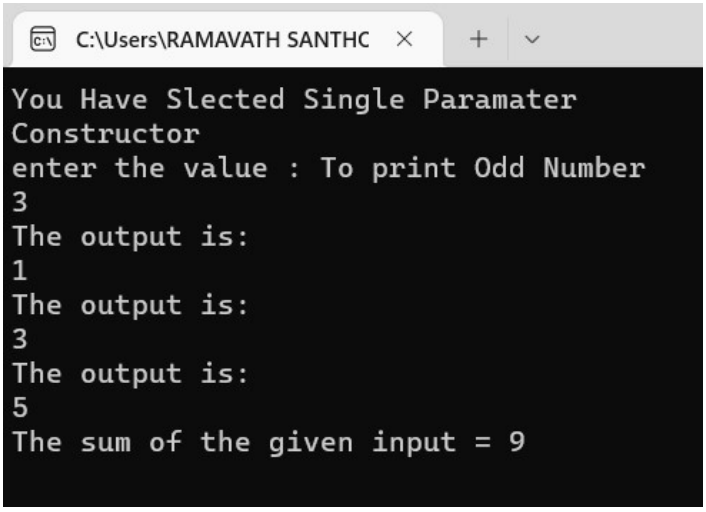



```
/*
Q6) Write a program in C++ to print Floyd's Triangle by using the constructor destructor.
*/
#include<iostream>
using namespace std;
class Floyds_Triangle
{
   public:
   int i,n,j,p = 1;
   //constructor
   Floyds_Triangle()
   {
     cout<<"\n Enter the number of rows : ";
     cin>>n;
      for(i=1;i<=n;i++)
      {
        for(j=1;j<=i;j++)
         {
          cout<< p << " ";
          p++;
         }
         cout << endl;
      }
   }
   ~Floyds_Triangle() // destructor
   {
   cout << "\n**********Floyds Triangle Printed**********";
   }
};
int main()
{
   Floyds_Triangle obj;
   return 0;
}
```

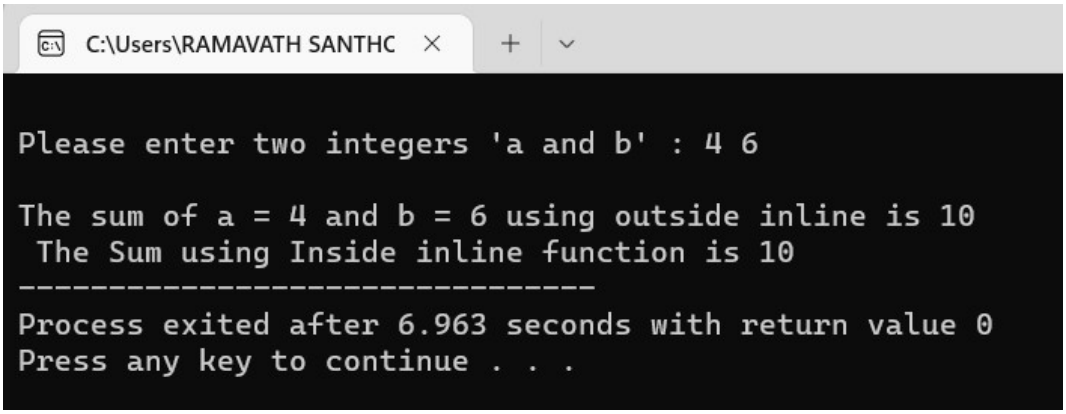



```
/*
Q7) Let's see the C++ program to show the Sum of n number of odd natural numbers by
using the different type of constructors.
*/
#include<iostream>
using namespace std;
class Sum_Constructor
{
private :
int i,n,num1 ,num2;
public :
Sum_Constructor(int a)
{
n=a;
int sum=0;
for(int i=1;i<=n;i++)
{
cout<<"The output is:"<<endl;
cout<<2*i-1<<" ";
sum=sum+(2*i-1);
cout<<endl;
}
cout<<"The sum of the given input = "<<sum<<endl;
}
Sum_Constructor(int a,int b)
{
cout<<"For 1st Input Answer is Given below"<<endl;
num1 = a;
int sum1=0;
for(int i=1;i<=num1 ;i++)
{
cout<<"The output is:";
cout<<2*i-1<<" ";
sum1=sum1+(2*i-1);
cout<<endl;
}
cout<<"The sum of the given input = "<<sum1<<endl;
cout<<endl;
```

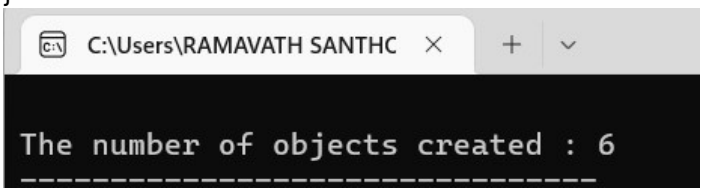
```
cout<<"For 2nd Input Answer is Given below."<<endl;
num2=b;
int sum2=0;
for(int i=1;i<=num2;i++)
{
cout<<"The output is:";
cout<<2*i-1<<" ";
sum2=sum2+(2*i-1);
cout<<endl;
}
cout<<"The sum of the given input = "<<sum2<<endl;
}
};
int main()
{
int ch;
cout<<"Enter 1 of Single parameter constructor."<<endl;
cout<<"Enter 2 For Multiple Paramter constucor."<<endl;
cout<<"Input 1 or 2 here : "<<endl;
cin>>ch;
system("cls");
if(ch == 1)
{
cout<<"You Have Slected Single Paramater"<<endl;
cout<<"Constructor"<<endl;
int n;
cout<<"enter the value : To print Odd Number"<<endl;
cin>>n;
Sum_Constructor a(n);
}
else if(ch == 2)
{
cout<<"You Have slected Multiple Paramater"<<endl;
cout<<"Constructor"<<endl;
int a,b;
cout<<"enter the 1st value : To print Odd Number"<<endl;
cin>>a;
cout<<"enter the 2nd value : To print Odd Number"<<endl;
cin>>b;
Sum_Constructor obj(a,b);
}
else
cout<<"Wrong Input ";
}
```




```
/*
Q8) Write a program using inline function inside and outside of the class (accessing data members
with objects and member functions)
*/
#include<iostream>
using namespace std;
class Sum
{
private :
int a,b;
public :
inline void add1(int c,int d)
{
a = c; b = d; cout << "\n The Sum using Inside inline function is " << a + b;
}
inline void add(int, int);
};
void Sum :: add(int x, int y)
{
a = x;
b = y;
cout << "\nThe sum of a = " << a << " and b = " << b << " using outside inline is " << a + b;
}
int main()
{
Sum o1;
int a, b;
cout << "\nPlease enter two integers 'a and b' : ";
cin >> a >> b;
o1.add(a,b);
o1.add1(a,b);
return 0;
}
```




```
/*
Q9) Write a program to demonstrate the use of static data members
*/
#include<iostream>
using namespace std;
// suppose we want to count the number of objects created
class Status
{
public :
static int count; // only once created and common for all objects
Status()
{
count++;
}
};
int Status :: count = 0;
int main()
{
Status s1,s2,s3,s4,s5,s6;
cout << "\nThe number of objects created : " << s1.count;
return 0;
}
```



/*
An EMPLOYEE class is to contain the following data members and member functions: Data members: EmployeeNumber (an integer), EmployeeName (a string of characters), BasicSalary (an integer), All Allowances (an integer), IT (an integer), NetSalary (aninteger). Member functions: to read the data of an employee, to calculate Net Salary and to print the values of

all the data members. (AllAllowances = 123% of Basic; Income Tax (IT) = 30% of the gross salary (=
basic Salary +AllAllowance); Net Salary = Basic Salary + All Allowances – IT)
(Write program using constructors, destructors, static data members and static Member functions)
*/

```
#include<iostream>
using namespace std;
class Employee
{
public :
static int EmployeeNumber;
string EmployeeName;
int BasicSalary;
int AllAllowances;
int IT;
int NetSalary;
int GrossSalary;
int n;
Employee()
{
fflush(stdin);
cin.clear();
EmployeeNumber++;
cout << "\nPlease Enter Employee Number " << EmployeeNumber << " details...";
cout << "\nPlease enter the Employee Name : ";
getline(cin,EmployeeName);
cout << "\nPlease enter the Basic Salary : ";
cin >> BasicSalary;
}
void calculation()
{
AllAllowances = 1.23 * BasicSalary;
GrossSalary = AllAllowances + BasicSalary;
IT = 0.3 * GrossSalary;
NetSalary = BasicSalary + AllAllowances - IT;
}
void showDetails()
{
cout << "\nEmployee Name : " << EmployeeName;
cout << "\nBasic Salary : " << BasicSalary;
cout << "\nGross Salary : " << GrossSalary;
cout << "\nNet Salary : " << NetSalary;
cout << "\nIncome Tax : " << IT;
}
~Employee()
{
cout << "\nEmployee " << n << "'s Details Done";
}
};
int Employee :: EmployeeNumber = 0;
```

```
int main()
{
int n;
cout << "\nPlease enter the number of Employees : ";
cin >> n;
Employee E[n];
for(int i = 0; i < n; i++)
{
//E[i].getDetails();
E[i].n = i + 1;
E[i].calculation();
}
for(int j = 0; j < n ; j++)
{
cout << "\nEmployee Number : " << j + 1;
E[j].showDetails();
cout << endl;
}
}
```

**********END**********