# Object-Oriented Programming System(OOPs) Lab

B-Tech Computer engineering

Semester - 5th

Code : CEN-591


Submitted By:-

Name :- Habiburrahman

Roll no :- 17BCS071

Submitted To:-

Mr. Jawahar Lal

Mr.Sarfaraz Masood

Department Of Computer Engineering And Technology ,

Jamia Millia Islamia ,New Delhi


Session : 2019-2020

Write a program in c++ to identify roots of Quadratic Equation .

```
#include<iostream>
#include<math.h>
using namespace std ;

void hrkhan()
{
        cout<<"Name :- Habiburrahman"<<endl;
        cout<<"Rollno :-  17BCS071 "<<endl;
        cout<<"B-Tech 5th Semester (Compute Engineering ) "<<endl;
        cout<<endl;
}

int main()
{
        hrkhan();
   float a , b,c ;
   cout<<"Enter the value of a  , b , c "<<endl ;
   cin>>a>>b>>c;
   float d=(b*b - (4*a*c));
   if (d==0)
   {
         float r = -(b/(2*a)) ;
         cout<<"Roots are Real and Equal "<< r <<" "<<r<<endl ;
   }
   else if(d>0)
   {
    float r1 = ((-b/(2*a)) - (sqrt(d)/(2*a))) ;
    float r2 = ((-b/(2*a)) + (sqrt(d)/(2*a))) ;
    cout<<"roots are real and Unequal  "<<r1 <<"     " <<r2<<endl;
   }
   else
   {
           float r1 = ((-b/(2*a))) ;
       float r2 = ((sqrt(-d)/(2*a))) ;
       cout<<"Roots are Imaginary   \n"<< r1 <<" + "<<r2<<" i "<<endl ;
       cout<< r1 <<" - "<<r2<<" i "<<endl;
   }

        return 0;
}
```

```
[Habiburrahman-khans-MacBook-Pro:~/desktop/oops] habiburrahmankhan% g++ roots.cpp
[Habiburrahman-khans-MacBook-Pro:~/desktop/oops] habiburrahmankhan% ./a.out
Name :- Habiburrahman
Rollno :-  17BCS071
B-Tech 5th Semester (Compute Engineering )

Enter the value of a  , b , c
1 -5 6
roots are real and Unequal  2    3
[Habiburrahman-khans-MacBook-Pro:~/desktop/oops] habiburrahmankhan% g++ roots.cpp
[Habiburrahman-khans-MacBook-Pro:~/desktop/oops] habiburrahmankhan% ./a.out
Name :- Habiburrahman
Rollno :-  17BCS071
B-Tech 5th Semester (Compute Engineering )

Enter the value of a  , b , c
1 1 1
Roots are Imaginary
-0.5 + 0.866025 i
-0.5 - 0.866025 i
[Habiburrahman-khans-MacBook-Pro:~/desktop/oops] habiburrahmankhan%
```

Write a program in c++ to demonstrate Banking Operation using class Such as Deposit into account , Withdraw from account with min balance condition , balance inquiry ,Exit .

```
#include<iostream>
#include<string>
#include<vector>
using namespace std ;
#define li long int
li  con =100000001230 ;
int no_of_account =0;
li total_money_inbank = 0 ;
void hrkhan()
{
        cout<<"Name :- Habiburrahman"<<endl;
        cout<<"Rollno :-  17BCS071 "<<endl;
        cout<<"B-Tech 5th Semester (Compute Engineering ) "<<endl;
        cout<<endl;
}
class account
{
   li acc_no , balance;
   string name ;
   char type ;
```

```
public:
        li getBalance()
  {
       return this->balance ;
  }
  char getType()
  {
       return this->type;
  }
  li getacc_no()
  {
    return this->acc_no ;
  }
  string getname()
  {
       return this->name ;
  }
  void createAccount(string name , char type)
  {
        no_of_account++;
        this->name = name ;
        this->type = type ;
        this->balance = 1500 ;
        total_money_inbank+=1500 ;
    this->acc_no = con + no_of_account;
    cout<<"Your Account is Created  "<<endl<<"Account No  is
"<<getacc_no()<<endl;
    cout<<"the initial balance is  "<<getBalance()<<endl;
  }
  void display()
  {
       cout<<"The Name of the  Account holder is "<<getname()<<endl;
       cout<<"The   Account no is "<<getacc_no()<<endl;
       cout<<"the type of account is   "<<getType()<<endl ;
       cout<<"The available balance is   "<<getBalance()<<endl;
  }
  void withdrawmoney(li money)
  {
        if (this->balance > money)
        {
              balance-=money;
              total_money_inbank-=money ;
        }
        else
        {
```

```
                cout<<"Insufficient Balance "<<endl;
        }
    }
    void depositmoney(li money)
    {
        this->balance+=money;
        total_money_inbank+=money ;
    }
};
vector<account> v ;
void displayacc(li account)
{
        int i =0 ;
    for (i = 0; i <v.size(); i++)
    {
        li acc = v[i].getacc_no() ;
        if (account==acc)
        {
            break ;
        }
    }
    if (i==v.size())
    {
        cout<<"Account no is Not Exist "<<endl;
    }
    else
    {
       v[i].display();
    }
}
 void withdraw(li acc , li money)
 {
     int i =0 ;
    for (i = 0; i <v.size(); i++)
    {
        li account = v[i].getacc_no() ;
        if (account==acc)
        {
            break ;
        }
    }
    if (i==v.size())
    {
        cout<<"Account no is Not Exist "<<endl;
    }
```

```
  else
  {
      v[i].withdrawmoney(money);
  }
 }
 void deposit(li acc , li money)
 {
    int i =0 ;
  for (i = 0; i <v.size(); i++)
  {
        li account = v[i].getacc_no() ;
        if (account==acc)
        {
              break ;
        }
  }
  if (i==v.size())
  {
       cout<<"Account no is Not Exist "<<endl;
  }
  else
  {
      v[i].depositmoney(money);
  }
 }
int main()
{
        hrkhan();
        int  c = 0 ;
        cout<<"1. Create Account \n2.Deposit \n 3. withdraw \n 4. display
Account detail \n5. All account detail\n5. total no of account in bank  \n 7.
Total money in Bank\n8. exit\n";
    while(1)
    {

      cin>>c ;
      switch(c)
      {
        case 1 :
              {
                 string name ;
                 char type ;
               cout<<"Enter the name "<<endl;
                    cin.ignore(10000,'\n');
                   cin.clear();
```

```
                getline(cin,name);
              cout<<"Enter the type of Account (Saving S or Current C )
"<<endl;
               cin>>type ;
         account a ;
          a.createAccount(name , type);
               v.push_back(a);
               break ;
               }
     case 2 :
              {
                li account ,money;
                cout<<"Enter the Account "<<endl;
                cin>>account ;
                cout<<"Enter the Money "<<endl;
                cin>>money;

                deposit(account , money);
                break;
              }
     case 3 :
                          {
                              li account ,money;
                cout<<"Enter the Account "<<endl;
                cin>>account ;
                cout<<"Enter the Money "<<endl;
                cin>>money;
                withdraw(account , money);
                break;
                          }
          case 4 :
                               {
               li acc;
               cout<<"Enter the Account "<<endl;
               cin>>acc ;
               displayacc(acc);
               break;
                               }
      case 5 :
            {
               for (int i = 0; i < v.size(); ++i)
               {
                      v[i].display();
               }
               break ;
```

```
            }
        case 6 :
                                {
                                 cout<<"total no of Account "<<no_of_account<<endl;
                                 break ;
                                }
            case 7 :
                    {
                        cout<<"Total no of Money   "<<total_money_inbank<<endl ;
                        break ;
                    }
             case 8 :
             {
                exit(0);
             }
             default :
             {
                cout<<"Invalid "<<endl;
             }
      }
    }
        return 0 ;
}
```

```
➜  oops g++ bankmanagementnot.cpp
➜  oops ./a.out
Name :- Habiburrahman
Rollno :-  17BCS071
B-Tech 5th Semester (Compute Engineering )

1. Create Account
2.Deposit
 3. withdraw
 4. display Account detail
5. All account detail
5. total no of account in bank
 7. Total money in Bank
8. exit
1
Enter the name
Anan
Enter the type of Account (Saving S or Current C )
S
Your Account is Created
Account No  is  100000001231
the initial balance is  1500
1
Enter the name
habib
Enter the type of Account (Saving S or Current C )
S
Your Account is Created
Account No  is  100000001232
the initial balance is  1500
2
Enter the Account
100000001232
Enter the Money
2500
3
Enter the Account
100000001232
Enter the Money
1500
4
Enter the Account
100000001232
The Name of the  Account holder is habib
The   Account no is 100000001232
the type of account is   S
The available balance is   2500
5
The Name of the  Account holder is Anan
The   Account no is 100000001231
the type of account is   S
The available balance is   1500
The Name of the  Account holder is habib
The   Account no is 100000001232
the type of account is   S
The available balance is   2500
6
total no of Account 2
7
Total no of Money   4000
8
➜  oops
```

Write a program in c++ to implement class and Constructor in use on Banking Operation as Deposit into account , Withdraw from account with min balance condition , balance inquiry ,Exit .

```
#include<iostream>
#include<string>
#include<vector>
using namespace std ;
#define li long int
li  con =100000001230 ;
int no_of_account =0;
li total_money_inbank = 0 ;
void hrkhan()
{
        cout<<"Name :- Habiburrahman"<<endl;
        cout<<"Rollno :-  17BCS071 "<<endl;
        cout<<"B-Tech 5th Semester (Compute Engineering ) "<<endl;
        cout<<endl;
}
class account
{
   li acc_no , balance;
   string name ;
   char type ;
public:
         li getBalance()
   {
        return this->balance ;
   }
   char getType()
   {
        return this->type;
   }
   li getacc_no()
   {
     return this->acc_no ;
   }
   string getname()
   {
        return this->name ;
   }
   account(string name , char type)
```

```
  {
        no_of_account++;
        this->name = name ;
        this->type = type ;
        this->balance = 1500 ;
        total_money_inbank+=1500 ;
    this->acc_no = con + no_of_account;
    cout<<"Your Account is Created  "<<endl<<"Account No  is
"<<getacc_no()<<endl;
    cout<<"the initial balance is  "<<getBalance()<<endl;
  }
  void display()
  {
       cout<<"The Name of the  Account holder is "<<getname()<<endl;
       cout<<"The   Account no is "<<getacc_no()<<endl;
       cout<<"the type of account is   "<<getType()<<endl ;
       cout<<"The available balance is   "<<getBalance()<<endl;
  }
  void withdrawmoney(li money)
  {
        if (this->balance > money)
        {
              balance-=money;
              total_money_inbank-=money ;
        }
        else
        {
              cout<<"Insufficient Balance "<<endl;
        }
  }
  void depositmoney(li money)
  {
        this->balance+=money;
        total_money_inbank+=money ;
  }
};
vector<account> v ;
void displayacc(li account)
{
       int i =0 ;
  for (i = 0; i <v.size(); i++)
  {
        li acc = v[i].getacc_no() ;
        if (account==acc)
        {
```

```
                break ;
            }
    }
    if (i==v.size())
    {
         cout<<"Account no is Not Exist "<<endl;
    }
    else
    {
        v[i].display();
    }
}
 void withdraw(li acc , li money)
 {
    int i =0 ;
    for (i = 0; i <v.size(); i++)
    {
            li account = v[i].getacc_no() ;
            if (account==acc)
            {
                break ;
            }
    }
    if (i==v.size())
    {
         cout<<"Account no is Not Exist "<<endl;
    }
    else
    {
        v[i].withdrawmoney(money);
    }
 }
 void deposit(li acc , li money)
 {
    int i =0 ;
    for (i = 0; i <v.size(); i++)
    {
            li account = v[i].getacc_no() ;
            if (account==acc)
            {
                break ;
            }
    }
    if (i==v.size())
    {
```

```
        cout<<"Account no is Not Exist "<<endl;
   }
   else
   {
       v[i].depositmoney(money);
   }
 }
int main()
{
        hrkhan();
        int  c = 0 ;
        cout<<"1. Create Account \n2.Deposit \n 3. withdraw \n 4. display
Account detail \n5. All account detail\n5. total no of account in bank  \n 7.
Total money in Bank\n8. exit\n";
    while(1)
    {

      cin>>c ;
      switch(c)
      {
         case 1 :
                {
                   string name ;
                   char type ;
                 cout<<"Enter the name "<<endl;
                      cin.ignore(10000,'\n');
                     cin.clear();
                     getline(cin,name);
                cout<<"Enter the type of Account (Saving S or Current C )
"<<endl;
                 cin>>type ;
                 account a(name , type);
                 v.push_back(a);
                 break ;
                 }
      case 2 :
              {
                 li account ,money;
                 cout<<"Enter the Account "<<endl;
                 cin>>account ;
                 cout<<"Enter the Money "<<endl;
                 cin>>money;
                 deposit(account , money);
                 break;
              }
```

```
case 3 :
                    {
                        li account ,money;
            cout<<"Enter the Account "<<endl;
            cin>>account ;
            cout<<"Enter the Money "<<endl;
            cin>>money;
            withdraw(account , money);
            break;
                    }
     case 4 :
                        {
            li acc;
            cout<<"Enter the Account "<<endl;
            cin>>acc ;
            displayacc(acc);
            break;
                        }
 case 5 :
       {
          for (int i = 0; i < v.size(); ++i)
          {
                v[i].display();
          }
          break ;
       }
  case 6 :
                      {
                       cout<<"total no of Account "<<no_of_account<<endl;
                       break ;
                      }
       case 7 :
             {
                  cout<<"Total no of Money   "<<total_money_inbank<<endl ;
                  break ;
             }
        case 8 :
        {
           exit(0);
        }
        default :
        {
           cout<<"Invalid "<<endl;
        }
}
```

```
    }
        return 0 ;
}
```

```
➜  oops g++ bankmanagement.cpp
➜  oops ./a.out
Name :- Habiburrahman
Rollno :-  17BCS071
B-Tech 5th Semester (Compute Engineering )

1. Create Account
2.Deposit
 3. withdraw
 4. display Account detail
5. All account detail
5. total no of account in bank
 7. Total money in Bank
8. exit
1
Enter the name
habib
Enter the type of Account (Saving S or Current C )
S
Your Account is Created
Account No  is  100000001231
the initial balance is  1500
1
Enter the name
Adnan
Enter the type of Account (Saving S or Current C )
S
Your Account is Created
Account No  is  100000001232
the initial balance is  1500
1
Enter the name
nouman
Enter the type of Account (Saving S or Current C )
S
Your Account is Created
Account No  is  100000001233
the initial balance is  1500
2
Enter the Account
100000001232
Enter the Money
5000
3
Enter the Account
100000001231
Enter the Money
1000
5
The Name of the  Account holder is habib
The   Account no is 100000001231
the type of account is   S
The available balance is   500
The Name of the  Account holder is Adnan
The   Account no is 100000001232
the type of account is   S
The available balance is   6500
The Name of the  Account holder is nouman
The   Account no is 100000001233
the type of account is   S
The available balance is   1500
6
total no of Account 3
4
Enter the Account
100000001232
The Name of the  Account holder is Adnan
The   Account no is 100000001232
the type of account is   S
The available balance is   6500
7
Total no of Money   8500
8
➜  oops
```

Write a program in c++ to make use of Inline & Friend function box class having 3 attributes length , breath , height and calculate volume and surface area .

```
#include<iostream>
#include<string>
using namespace std ;

void hrkhan()
{
        cout<<"Name :- Habiburrahman"<<endl;
        cout<<"Rollno :-  17BCS071 "<<endl;
        cout<<"B-Tech 5th Semester (Compute Engineering ) "<<endl;
        cout<<endl;
}

class Box
{
        private :
        int length ;
        int breath ;
        int height ;
        public :
        inline void setlength(int length)
        {
       this->length = length ;
        }

        inline void setbreath(int breath)
        {
       this->breath= breath ;
        }
        inline void setheight(int height)
        {
                this->height = height ;
        }
        inline int getlength()
        {
                return this->length ;
        }

        inline int getbreath()
        {
```

```
            return this->breath ;
        }

        inline int getheight()
        {
            return this->height ;
        }
        Box()
        {

        }

        Box(int length , int breath , int height)
        {
            this->length = length ;
            this-> breath = breath ;
            this-> height = height ;
        }
    friend int volume(Box b);
    friend int surfacearea(Box b);
};
  int volume(Box b)
  {
     return b.length*b.breath*b.height;
  }
  int surfacearea(Box b)
  {
        return 2*(b.length*b.breath + b.breath*b.height + b.height*b.length) ;
  }
int main()
{
    Box box;
    hrkhan();
    int l , b ,h ;
    cout<<"Enter the length breath and height "<<endl ;
    cin>>l>>b>>h ;
    box.setlength(l);
    box.setheight(h);
    box.setbreath(b);
    cout<<" the length is "<<box.getlength()<<endl;
    cout<<" the breath is "<<box.getbreath()<<endl;
    cout<<" the height is "<<box.getheight()<<endl;
    cout<<" the volume is "<<volume(box)<<endl;
    cout<<"The surface area is "<<surfacearea(box)<<endl;
    cout<<" Using Constructor "<<endl ;
```

```
    Box box2(10,15,25);
    cout<<" the length is "<<box2.getlength()<<endl;
    cout<<" the breath is "<<box2.getbreath()<<endl;
    cout<<" the height is "<<box2.getheight()<<endl;
    cout<<" the volume is "<<volume(box2)<<endl;
    cout<<"The surface area is "<<surfacearea(box2)<<endl;
        return 0 ;
}
```

```
The surface area is 1550
➜  oops g++ Box.cpp
➜  oops ./a.out
Name :- Habiburrahman
Rollno :-  17BCS071
B-Tech 5th Semester (Compute Engineering )

Enter the length breath and height
5
5
5
 the length is 5
 the breath is 5
 the height is 5
 the volume is 125
The surface area is 150
 Using Constructor
 the length is 10
 the breath is 15
 the height is 25
 the volume is 3750
The surface area is 1550
➜  oops
```

Write a program in c++ to make use of concept of Polymorphism , Inheritance & Virtual function in finding the area of rectangle , triangle & circle of function base class Shape .

```
#include <iostream>
using namespace std;
void hrkhan()
{
    cout<<"Name :- Habiburrahman"<<endl;
    cout<<"Rollno :-  17BCS071 "<<endl;
    cout<<"B-Tech 5th Semester (Compute Engineering ) "<<endl;
    cout<<endl;
}
class Shape
{
protected:
    int width,height;
    float radius;
public:
    Shape(int a=0,int b=0)
    {
        width=a;
        height=b;
        radius=0;
    }
    Shape(float rad)
    {
        width=0;
        height=0;
        radius=rad;
    }
    virtual void area()
    {
        cout << "Parent class area: " << endl;
    }
};

class Rectangle:public Shape
{
    public:
    Rectangle(int a,int b):Shape(a,b)
    {

    }
    void area()
    {
        cout << "rectangle area: " << width*height;
    }
```

```
};

class Triangle:public Shape
{
    public:
    Triangle(int a,int b):Shape(a,b)
    {}
    void area()
    {
        cout << "\n\nTriangle class area: " << (float)(width*height/2);
    }
};

class Circle:public Shape
{
    public:
    Circle(float r):Shape(r)
    {

    }
    void area()
    {
        cout << "\n\nArea of circle is: " << (float)(3.1416*radius*radius);
    }
};
int main()
{
    hrkhan();
         Shape *shape;
         Rectangle rec(10,7);
         Triangle tri(10,5);
         Circle cir(10);
         shape=&rec;
         shape->area();
         shape=&tri;
         shape->area();
         shape=&cir;
         shape->area();
         cout << "\n\n";
         return 0;
}
```

```
[Habiburrahman-khans-MacBook-Pro :: ~/desktop » cd c++program
[Habiburrahman-khans-MacBook-Pro :: ~/desktop/c++program » g++ shape.cpp
[Habiburrahman-khans-MacBook-Pro :: ~/desktop/c++program » ./a.out
Name :- Habiburrahman
Rollno :-  17BCS071
B-Tech 5th Semester (Compute Engineering )

rectangle area: 70

Triangle class area: 25

Area of circle is: 314.16

Habiburrahman-khans-MacBook-Pro :: ~/desktop/c++program »
```

## Write a program in c++ to demonstrate use of operator overloading in unary operator and binary operator .

```
#include<iostream>
using namespace std ;
void hrkhan()
{
        cout<<"Name :- Habiburrahman"<<endl;
        cout<<"Rollno :-  17BCS071 "<<endl;
        cout<<"B-Tech 5th Semester (Compute Engineering ) "<<endl;
        cout<<endl;
}
class Complex
{
        private :
        int real ;
        int imaginary ;
        public :
        Complex()
        {

        }
        Complex(int real , int imaginary)
        {
                this->real = real;
                this->imaginary = imaginary;
        }
        Complex operator+(Complex &c)
```

```
    {
        Complex temp ;
        temp.real =this->real + c.real ;
        temp.imaginary =this->imaginary + c.imaginary ;
        return temp ;
    }

    Complex operator-(Complex &c)
    {
        Complex temp ;
        temp.real =this->real - c.real ;
        temp.imaginary =this->imaginary - c.imaginary ;
        return temp ;
    }
    Complex operator*(Complex &c)
    {
        Complex temp ;
        temp.real =this->real * c.real ;
        temp.imaginary =this->real * c.imaginary ;
        return temp ;
    }
void operator++()
{
    this->real++;
    this->imaginary++;
}
void operator--()
{
    this->real = this->real -1 ;
    this->imaginary = this->imaginary -1;
}
    friend istream & operator>>(istream &din , Complex &c)
    {
        din>>c.real ;
        din>>c.imaginary;
        return din ;
    }
    friend ostream & operator<<(ostream &dout , Complex &c)
    {
        dout<<c.real<<" ";
        if (c.imaginary > 0 )
        {
            dout<<"+";
        }
        dout<<c.imaginary<<"i "<<endl;
```

```
                return dout ;
        }
};

int main()
{
   Complex c1 , c2;
   hrkhan();
   cout<<"Enter the real and Complex part "<<endl ;
   cin>>c1;
    cout<<"Enter the real and Complex part "<<endl ;
   cin>>c2;
   Complex c3 ;
   cout<<c1<<endl;
   cout<<c2<<endl;
   c3 = c1 + c2 ;
   cout<<"Addition using binary operator overload      ";
   cout<<c3<<endl;
   c3 = c1 - c2 ;
   cout<<"Subtraction using binary operator overload       ";
   cout<<c3<<endl;
   c3 = c1 * c2 ;
   cout<<"Multiplication  using binary operator overload    ";
   cout<<c3<<endl ;

   ++c1;
   --c2;
    cout<<"Increment using unaray operator overload    ";
   cout<<c1<<endl;
    cout<<"decrement using unaray operator overload    ";
   cout<<c2<<endl;
        return 0 ;
}
```

```
➜  c++program g++ operatoroverloading.cpp
➜  c++program ./a.out
Name :- Habiburrahman
Rollno :-  17BCS071
B-Tech 5th Semester (Compute Engineering )

Enter the real and Complex part
10
15
Enter the real and Complex part
5
10
10 +15i

5 +10i

Addition using binary operator overload      15 +25i

Subtraction using binary operator overload       5 +5i

Multiplication  using binary operator overload    50 +100i

Increment using unaray operator overload    11 +16i

decrement using unaray operator overload    4 +9i

➜  c++program
```

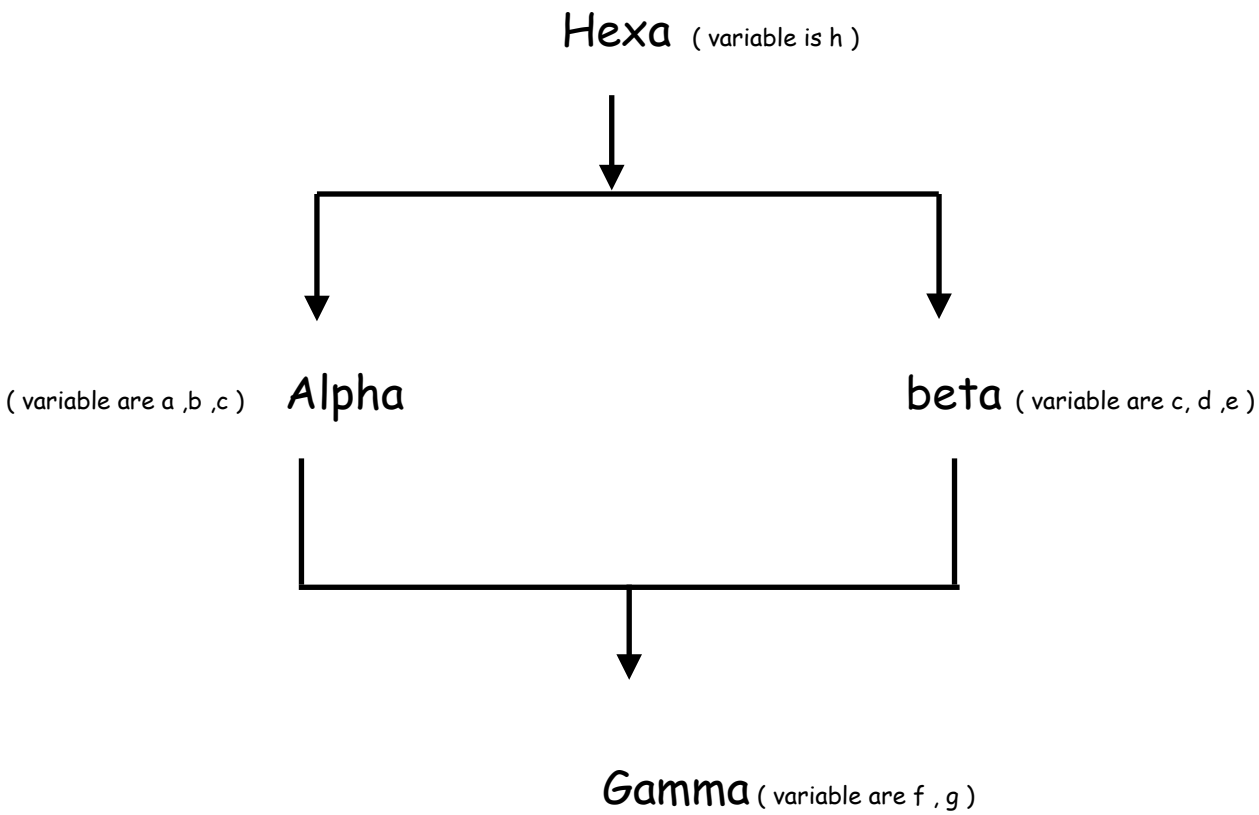
## Write a program in c++ to highlight the difference between overloaded assignment operator & copy Constructor .

```
#include<iostream>
using namespace std ;
void hrkhan()
{
        cout<<"Name :- Habiburrahman"<<endl;
        cout<<"Rollno :-  17BCS071 "<<endl;
        cout<<"B-Tech 5th Semester (Compute Engineering ) "<<endl;
        cout<<endl;
}

class Gamma
{
        int a , b ;
        public :
   Gamma()
```

```
    {

    }
    Gamma(int a , int b)
    {
        this->a = a ;
        this -> b = b ;
        cout<<"Constructor Called "<<endl;
    }
    Gamma(Gamma &g)
    {
        this->a=g.a ;
        this->b = g.b ;
        cout<<"Copy Constructor Called "<<endl;
    }
    void print()
    {
       cout<< a << "    "<<b<<endl ;
    }
    void operator=(Gamma &gamma)
    {
      this->a = gamma.a ;
      this->b = gamma.b;
      cout<<"Overloaded Assignment operator Called"<<endl;
    }
};

int main()
{
        hrkhan();
  Gamma g(1 ,2 );
  cout<<"Value of g are   ";
  g.print();
  Gamma g2 = g ;
  cout<<"Values of g2 are ";
  g2.print();
  Gamma g3(g);
  cout<<"Values of g3 are ";
  g3.print();
  Gamma g4 ;
  g4 = g ;
  cout<<"Values of g4 are ";
  g4.print();
        return 0 ;
}
```

```
Values of g4 are 1    2
Habiburrahman-khans-MacBook-Pro :: ~/desktop/c++program » g++ copyconstructor.cpp
Habiburrahman-khans-MacBook-Pro :: ~/desktop/c++program » ./a.out
Name :- Habiburrahman
Rollno :-  17BCS071
B-Tech 5th Semester (Compute Engineering )

Constructor Called
Value of g are   1    2
Copy Constructor Called
Values of g2 are 1    2
Copy Constructor Called
Values of g3 are 1    2
Overloaded Assignment operator Called
Values of g4 are 1    2
Habiburrahman-khans-MacBook-Pro :: ~/desktop/c++program »
```

Write a program in c++ to illustrate how the Constructors are implemented and the order in which they are called when the classes are inherited use 4 classes named hexa , alpha , beta , gamma , fig is as shown below



```
#include<iostream>
using namespace std ;
void hrkhan()
{
        cout<<"Name :- Habiburrahman"<<endl;
        cout<<"Rollno :-  17BCS071 "<<endl;
        cout<<"B-Tech 5th Semester (Compute Engineering ) "<<endl;
        cout<<endl;
}
class hexa
{
    int h ;
     public:
        int geth()
        {
                return this->h;
        }
        hexa(int h )
        {
                cout<<"Hexa Constructor is Called "<<endl;
                this->h = h ;
        }
        ~hexa()
        {
                cout<<"Hexa desctructor is Called "<<endl;
        }
};
class alpha : virtual public hexa
{
private:
   int a ,b,c;
   public:
   int geta()
   {
        return this->a ;
   }
   int getb()
   {
        return this->b ;
   }
   int getc()
   {
        return this->c ;
   }
   alpha(int a , int b , int c , int h) : hexa(h)
```

```
  {
        cout<<"Alpha Constructor is called "<<endl;
        this->a = a ;
        this->b = b ;
        this->c = c ;
  }
  ~alpha()
  {
        cout<<"Alpha Destructor is Called "<<endl ;
  }
};
class beeta : virtual public hexa
{
 private:
        int c , d,e ;
 public:
        int getcbeeta()
        {
                return this->c;
        }
        int getd()
        {
                return this->d;
        }
                int gete()
        {
                return this->e;
        }
        beeta(int c ,int d,int e , int h) : hexa(h)
        {
                cout<<"Beeta constructor is Called"<<endl;
                this->c = c ;
                this->d = d ;
                this->e = e ;
        }
        ~beeta()
        {
                cout<<"Beeta Destructor is Called "<<endl;
        }
};
class gamma  : virtual public alpha  , virtual public beeta
{
  int f , g ;
public:
                int getf()
```

```
        {
                return this->f;
        }
                int getg()
        {
                return this->g;
        }
        gamma(int f , int g , int a , int b , int c , int d , int e , int h ) : alpha(a,b,c ,
h) , beeta(c , d , e , h) , hexa(h)
        {
                cout<<"Gamma Constructor is Called"<<endl;
                this->f = f ;
                this->g = g ;
        }
        ~gamma()
        {
                cout<<"Gamma Constructor is called"<<endl ;
        }
        void print()
        {
                cout<<"In Alpha class "<<endl;
                cout<<"a "<<geta()<<endl;
                cout<<"b "<<getb()<<endl;
                cout<<"c "<<getc()<<endl;
                cout<<"In beeta class "<<endl;
                cout<<"c "<<getcbeeta()<<endl;
                cout<<"d "<<getd()<<endl;
                cout<<"e "<<gete()<<endl;
                cout<<"In gamma class "<<endl;
                cout<<"f "<<getf()<<endl;
                cout<<"g "<<getg()<<endl;
                cout<<"In hexa class "<<endl;
                cout<<"h "<<geth()<<endl;

        }
};


int main()
{
        hrkhan();
    gamma g(1,2,3,4,5,6,7,8);
    g.print();
        return 0 ;
}
```

```
Habiburrahman-khans-MacBook-Pro :: ~/desktop/c++program » g++ constructorordercalled.cpp
Habiburrahman-khans-MacBook-Pro :: ~/desktop/c++program » ./a.out
Name :- Habiburrahman
Rollno :-  17BCS071
B-Tech 5th Semester (Compute Engineering )

Hexa Constructor is Called
Alpha Constructor is called
Beeta constructor is Called
Gamma Constructor is Called
In Alpha class
a 3
b 4
c 5
In beeta class
c 5
d 6
e 7
In gamma class
f 1
g 2
In hexa class
h 8
Gamma Constructor is called
Beeta Destructor is Called
Alpha Destructor is Called
Hexa desctructor is Called
Habiburrahman-khans-MacBook-Pro :: ~/desktop/c++program »
```

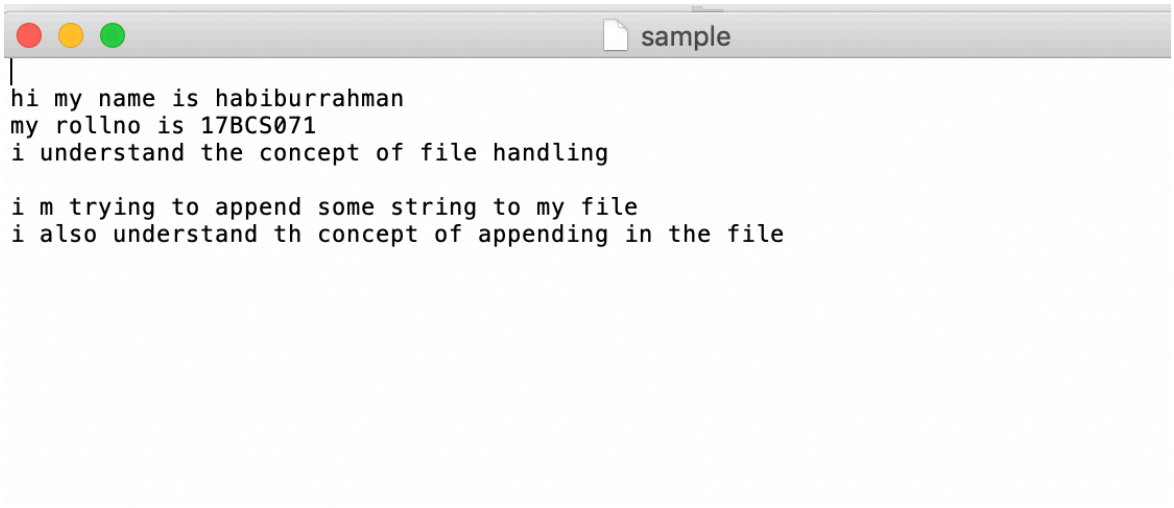
## Write a program to implement following operation in file handing in c++  read from a file , write from a file , append from a file .

```
#include<iostream>
#include<fstream>
#include<string>
using namespace std ;

void hrkhan()
{
        cout<<"Name :- Habiburrahman"<<endl;
        cout<<"Rollno :-  17BCS071 "<<endl;
        cout<<"B-Tech 5th Semester (Compute Engineering ) "<<endl;
        cout<<endl;
}
void createfile()
{
```

```
        cout<<"Enter the name of file "<<endl;
        string name ;
        cin>>name ;
        ofstream fout ;
        fout.open(name);
        string line ;
        cout<<"Enter -1 to exit "<<endl ;
        while(fout)
        {
                getline(cin , line);
                if(line=="-1")
                {
                        break ;
                }
                fout<<line<<endl ;
        }
        cout<<"file is Created Successfully ...."<<endl ;
}
 void readfile()
 {
        cout<<"Enter the name of file  "<<endl;
        string name ;
        cin>>name ;
        ifstream fin ;
        fin.open(name);
        string line ;
        while(fin)
        {
                getline(fin, line );
                cout<<line<<endl ;
        }
        fin.close();
 }
void appendfile()
{
        cout<<"Enter the name of file "<<endl ;
        string name ;
        cin>>name;
        ofstream fout ;
        fout.open(name , ios::beg | ios::app) ;
        string line ;
        cout<<"Enter -1 to exit "<<endl ;
        while(fout)
        {
                getline(cin , line);
```

```
            if(line=="-1")
            {
                  break ;
            }
            fout<<line<<endl ;
      }
      cout<<"file is Successfully Updated  ...."<<endl ;
}
int main()
{
      hrkhan();
      while(1)
      {

            cout << "1. Write to createfile  ( or  overwrite )\n2. Write to
file( append )\n3. Read File\n 4. Exit\nEnter choice : ";
            int option;
            cin >> option;
            switch(option)
            {
                  case 1:
                              createfile();
                              break;
                  case 2:
                              appendfile();
                              break;
                  case 3:
                              readfile();
                              break;
                  case 4:
                              cout << "Exit \n";
                              exit(0);
                  default:
                              cout << "Invalid option please Enter Again \n";
                              break;
            }
      }

      return 0 ;
}
```

```
Habiburrahman-khans-MacBook-Pro :: ~/desktop/c++program » g++ filehandling1.cpp
Habiburrahman-khans-MacBook-Pro :: ~/desktop/c++program » ./a.out
Name :- Habiburrahman
Rollno :-  17BCS071
B-Tech 5th Semester (Compute Engineering )

1. Write to createfile  ( or  overwrite )
2. Write to file( append )
3. Read File
 4. Exit
Enter choice : 1
Enter the name of file
sample
Enter -1 to exit
hi my name is habiburrahman
my rollno is 17BCS071
i understand the concept of file handling
-1
file is Created Successfully ....
1. Write to createfile  ( or  overwrite )
2. Write to file( append )
3. Read File
 4. Exit
Enter choice : 3
Enter the name of file
sample

hi my name is habiburrahman
my rollno is 17BCS071
i understand the concept of file handling

1. Write to createfile  ( or  overwrite )
2. Write to file( append )
3. Read File
 4. Exit
Enter choice : 2
Enter the name of file
sample
Enter -1 to exit
i m trying to append some string to my file
i also understand th concept of appending in the file
-1
file is Successfully Updated  ....
1. Write to createfile  ( or  overwrite )
2. Write to file( append )
3. Read File
 4. Exit
Enter choice : 3
Enter the name of file
sample

hi my name is habiburrahman
my rollno is 17BCS071
i understand the concept of file handling

i m trying to append some string to my file
i also understand th concept of appending in the file

1. Write to createfile  ( or  overwrite )
2. Write to file( append )
3. Read File
 4. Exit
Enter choice : 4
Exit
Habiburrahman-khans-MacBook-Pro :: ~/desktop/c++program »
```




Write a program to implement following operation in file handing in c++ Copy one file to another , copy only alpha numeric , read from a specific position .

```
#include<iostream>
#include<fstream>
#include<cstring>
using namespace std ;
void hrkhan()
{
        cout<<"Name :- Habiburrahman"<<endl;
        cout<<"Rollno :-  17BCS071 "<<endl;
        cout<<"B-Tech 5th Semester (Compute Engineering ) "<<endl;
        cout<<endl;
}

bool checkword(string str)
{
        int flagalpha = 0 , flagnumeric = 0 ;
   for (int i = 0; i < str.size(); ++i)
   {
      if(isalpha(str[i]))
      {
         flagalpha = 1;
      }
      else if(isdigit(str[i]))
      {
       flagnumeric = 1 ;
      }
      else
```

```
        {
         flagalpha = 0 ;
         flagnumeric = 0 ;
         break ;
        }
    }
    if(flagalpha==1 && flagnumeric ==1)
    {
        return 1 ;
    }
    else{
        return 0 ;
    }

    return 1 ;
}
void copyfile()
{
        cout<<"Enter the name of file to copy "<<endl;
        string namesource ;
        cin>>namesource;
        cout<<"Enter the name of to copy the content  "<<endl ;
        string namedestination ;
        cin>>namedestination ;
        ofstream fout ;
        ifstream fin ;
        fin.open(namesource);
        fout.open(namedestination);
        string line ;
        while(fin)
        {
                getline(fin, line);
                fout<<line<<endl ;
        }
        fin.close();
        fout.close();
}
void readSpecific()
{
        cout<<"Enter the name of file "<<endl ;
        string name ;
        cin>>name ;
        ifstream fin ;
        fin.open(name);
        cout<<" enter the index from where to display "<<endl;
```

```
        int start ;
        cin>>start ;
        fin.seekg(start-1,ios::beg);
        char c ;
        cout<<"Enter the end index  from where to display "<<endl;
        int end ;
        cin>>end ;
        end = end - start ;
        int i = 0 ;
        while(i < end && fin)
        {
                fin>>noskipws>>c ;
                cout<<c;
                i++ ;
        }
        cout<<endl ;
}
void readAlphaNum()
{
        cout << "Enter name of file: "<<endl;
    string name;
    cin >> name;
    ifstream fin;
    fin.open(name);
    string word;
    while(fin >> word)
    {
        if(checkword(word))
        {
          cout << word <<endl;
        }
    }
}

int main()
{
    hrkhan();
     int choice;
    while(1)
    {
        cout << "1. Copy one file to another\n2. Copy only alphanumeric\n3.
Read from specific positon\n4. Exit\nEnter  choice: ";
        cin >> choice;
        switch(choice)
        {
```

```
            case 1:
                        copyfile();
                        break;
            case 2:
                        readAlphaNum();
                        break;
            case 3:
                        readSpecific();
                        break;
            case 4:
                        cout  << "Exit\n";
                        exit(0);
            default:
                        cout << "Please enter a valid choice\n";
        }
    }


        return 0 ;
}
```

```
Habiburrahman-khans-MacBook-Pro :: ~/desktop/c++program » g++ filehandling2.cpp
Habiburrahman-khans-MacBook-Pro :: ~/desktop/c++program » ./a.out
Name :- Habiburrahman
Rollno :-  17BCS071
B-Tech 5th Semester (Compute Engineering )

1. Copy one file to another
2. Copy only alphanumeric
3. Read from specific positon
4. Exit
Enter  choice: 1
Enter the name of file to copy
sample
Enter the name of to copy the content
sample2
1. Copy one file to another
2. Copy only alphanumeric
3. Read from specific positon
4. Exit
Enter  choice: 3
Enter the name of file
sample2
 enter the index from where to display
1
Enter the end index  from where to display
1000

hi my name is habiburrahman
my rollno is 17BCS071
i understand the concept of file handling

i m trying to append some string to my file
i also understand th concept of appending in the file



1. Copy one file to another
2. Copy only alphanumeric
3. Read from specific positon
4. Exit
Enter  choice: 3
Enter the name of file
sample
 enter the index from where to display
5
Enter the end index  from where to display
20
my name is habi
1. Copy one file to another
2. Copy only alphanumeric
3. Read from specific positon
4. Exit
Enter  choice: 2
Enter name of file:
file1
17BCS071
HNO51
rfgrf3433
1. Copy one file to another
2. Copy only alphanumeric
3. Read from specific positon
4. Exit
Enter  choice: 4
Exit
Habiburrahman-khans-MacBook-Pro :: ~/desktop/c++program »
```

sample

```
hi my name is habiburrahman
my rollno is 17BCS071
i understand the concept of file handling

i m trying to append some string to my file
i also understand th concept of appending in the file
```

sample2

```
hi my name is habiburrahman
my rollno is 17BCS071
i understand the concept of file handling

i m trying to append some string to my file
i also understand th concept of appending in the file
```

file1

```
my name is habiburrahman
rollno 17BCS071
HN051 noor nagar 1-D
b-tech computer 12121212233
43433 rfgrf3433
```

Write a program in c++ to implement Mathematical operation on class (string ) using operator overloading .

```
#include<iostream>
#include<cstring>
using namespace std;
void hrkhan()
{
        cout<<"Name :- Habiburrahman"<<endl;
        cout<<"Rollno :-  17BCS071 "<<endl;
        cout<<"B-Tech 5th Semester (Compute Engineering ) "<<endl;
        cout<<endl;
}
class String
{
```

```
    char *p;
    int len ;
public:
        String()
        {

        }

    String(char *s)
    {
      int len = length(s);
           p = new char[len + 1];
            int i=0;
            while(*(s+i)!='\0')
            {
                  *(p+i)=*(s+i);
                  i++;
            }
            *(p+i)='\0';

    }
        int leng()
        {
            return length(this->p);
        }
    void showdata()
    {

       cout<<p<<endl;
    }
     int length(char *s)
    {
        int i=0;
        while(*(s+i)!='\0')
          {

                i++;
          }
         return i;
        }
        void operator==(String &a)
        {
                int len = length(a.p);
          this->p = new char[len + 1];
           int i=0;
```

```
   while(*(a.p+i)!='\0')
   {
        *(p+i)=*(a.p+i);
        i++;
   }
   *(p+i)='\0';
}
 friend ostream & operator<< (ostream &dout,String &a)
 {
        dout<<a.p;
        return dout;
 }
        int operator<=(String &a)
{
        int len1 =a.len;
        int len2 =this->len;
   int i=0 , j=0;
   while(i<len1 && j<len2)
   {
      if(*(a.p+i)!=*(this->p +j))
      {
        return 0;
          }
        i++;
        j++;
   }
  if(i==j)
  {
        return 1;
  }
  else
  {
        return 0;
  }

}
String operator+(String &a)
{
        String temp ;
        temp.len = length(this->p) + length(a.p);
         temp.p = new char[temp.len + 1];
         int i=0;
         while(*(this->p+i)!='\0')
   {
        *(temp.p+i)=*(this->p+i);
```

```
            i++;
        }
        int j=0;
         while(*(a.p+j)!='\0')
        {
            *(temp.p+i)=*(a.p+j);
            j++;
            i++;
        }
        *(p+i)='\0';
       return temp;
      }

};
int main()
{
      hrkhan();
      char arr[] = "hello" ;
String str1(arr);
cout<<"The  String 1   : ";
str1.showdata();
cout<<"The length of a String 1   :  "<<str1.leng()<<endl;
char arr1[] = "world";
String str2(arr1);
cout<<"The  String 2   : ";
str2.showdata();
cout<<"The length of a String 2   :  "<<str2.leng()<<endl;
String str3;
str3 = str1 + str2 ;
cout<<"The String 3   : ";
str3.showdata();
cout<<"The length of a String after addition with + overload operator
"<<str3.leng()<<endl;
String str4;
str4 == str3;
cout<<"The  String 4   : ";
str4.showdata();
cout<<"The length of a String after  == overload operator operation perform
"<<str4.leng()<<endl;
int m = str4 <= str1;
cout<<"Is String 4 is <= String 3    (1 for true 0 for false ):   "<<m<<endl;
cout<<"  operator Overload << operation to print String 1   :  " ;
cout<<str1<<endl;
   return 0;
}
```

```
Habiburrahman-khans-MacBook-Pro :: ~/desktop/c++program » g++ string.cpp
Habiburrahman-khans-MacBook-Pro :: ~/desktop/c++program » ./a.out
Name :- Habiburrahman
Rollno :-  17BCS071
B-Tech 5th Semester (Compute Engineering )

The  String 1   : hello
The length of a String 1   :  5
The  String 2   : world
The length of a String 2   :  5
The String 3   : helloworld
The length of a String after addition with + overload operator   10
The  String 4   : helloworld
The length of a String after  == overload operator operation perform   10
Is String 4 is <= String 3    (1 for true 0 for false ):   1
  operator Overload << operation to print String 1   :  hello
Habiburrahman-khans-MacBook-Pro :: ~/desktop/c++program »
```

Write a program to implement basic linked list operation in c++ as create linked list , insert at beginning , delete at beginning , insert node at location i , delete at location i ,insert at end , delete at end , display the content , calculate the length of linked list .

```
#include<iostream>
using namespace std ;


void hrkhan()
{
	cout<<"Name :- Habiburrahman"<<endl;
	cout<<"Rollno :-  17BCS071 "<<endl;
	cout<<"B-Tech 5th Semester (Compute Engineering ) "<<endl;
	cout<<endl;
}

template<class T>
class Node
{
	public :
```

```
   T data ;
   Node *next ;
   Node(T data)
   {
     this->data= data;
     this->next=NULL ;
   }

};
template <class T>
class linkedlist
{
        public :
        Node<T>* head ;
   Node<T> *tail ;
        int size ;
        void createlist()
        {
                this->size=0;
                this->head=NULL;
                this->tail= NULL;
        }
        void insertatbegin(T data);
        void insertatlast(T data);
        void insertatindex(T data , int i );
        T deleteatbegin();
        T deleteatlast();
        T deleteatindex(int i);
        void display();
        int length();
};
template<class T >
void linkedlist<T> :: insertatbegin(T data)
{
        Node<T> *node = new Node<T>(data);
  if (head==NULL)
  {
        size++;
      head = node ;
      tail = node ;
      return;
  }
  node->next = head;
  head = node ;
  size++;
```

```
}
template<class T>
T linkedlist<T> :: deleteatbegin()
{
  if (head==NULL)
  {
    cout<<"the linked list is empty"<<endl ;
    return 0 ;
  }
  T data = head->data ;
  if (head==tail)
   {
    head=head->next;
    tail=head ;
    size--;
    return data ;
    }
  head = head->next ;
  size--;
  return data ;
}

template<class T >
void linkedlist<T> :: insertatlast(T data)
{
        Node<T> *node = new Node<T>(data);
  if (head==NULL)
  {
        size++;
     head = node ;
     tail = node ;
     return;
  }
  tail->next = node;
  tail = node ;
  size++;
}
template<class T>
T linkedlist<T> :: deleteatlast()
{
  if (head==NULL)
  {
    cout<<"the linked list is empty"<<endl ;
    return 0 ;
  }
```

```
  Node<T> *temp = head ;
  T last = tail->data ;
  while(temp->next!=tail)
  {
    temp=temp->next ;
  }
  tail = temp ;
  temp->next = NULL ;
   size--;
   return last ;
}
template<class T >
void linkedlist<T> :: display()
{
        Node<T> *node = head ;
        while(node!=NULL)
        {
                cout<< node->data <<endl;
                node= node->next ;
        }
        cout<<endl;
}
template<class T>
int linkedlist<T> :: length()
{
  return this->size ;
}

template<class T>
void linkedlist<T> :: insertatindex(T data , int i)
{

  if (i>this->size || i==0)
  {
    cout<<"th lenth of linked list is less "<<endl;
    return;
  }
  if (i==1)
  {
     insertatbegin(data);
  }else if (i==this->size)
  {
    insertatlast(data);
  }
  else
```

```
  {
    Node<T> *node = head ;
    for (int j = 0; j<i-2; j++)
    {
        node=node->next ;
    }
    Node<T> *newNode = new Node<T>(data);
    newNode->next = (node->next);
    node->next = newNode ;
     this->size++;
  }

}
template<class T>
T linkedlist<T> :: deleteatindex(int i)
{
  T data=0;
  if (i>this->size || i==0)
  {
    cout<<"th lenth of linked list is less "<<endl;
  }
  if (i==1)
  {
     return deleteatbegin();
  }else if (i==this->size)
  {
   return deleteatlast();
  }
  else
  {
        Node<T> *node = head ;
    for (int j = 0; j<i-2; j++)
    {
        node=node->next ;
    }
    data =node->next->data ;
    node->next=node->next->next;
     this->size--;
     return data ;
  }

}

int main()
{
```

```
        hrkhan();
    linkedlist<int> ll ;
    ll.createlist();
    ll.insertatbegin(10);
    ll.insertatlast(20);
    ll.insertatlast(45);
    ll.insertatlast(55);
    ll.insertatlast(65);
    ll.insertatbegin(5);
    ll.insertatlast(25);
    ll.insertatindex(15 ,3);
    ll.display();
     cout<<"the length ofthe list "  << ll.length()<<endl;
    cout<<"the deleted element is from begin " <<ll.deleteatbegin()<<endl;
    cout<<"the deleted element is from end " <<ll.deleteatlast()<<endl;
    cout<<"the deleted element at positon i  " <<ll.deleteatindex(2)<<endl;
    ll.display();
     cout<<"the length ofthe list "  << ll.length()<<endl;
        return 0 ;
}
```

```
Habiburrahman-khans-MacBook-Pro :: ~/desktop/c++program <master*> » g++ linkedlist.cpp
Habiburrahman-khans-MacBook-Pro :: ~/desktop/c++program <master*> » ./a.out
Name :- Habiburrahman
Rollno :-  17BCS071
B-Tech 5th Semester (Compute Engineering )

5
10
15
20
45
55
65
25

the length ofthe list 8
the deleted element is from begin 5
the deleted element is from end 25
the deleted element at positon i  15
10
20
45
55
65

the length ofthe list 5
Habiburrahman-khans-MacBook-Pro :: ~/desktop/c++program <master*> »
```