Department of Computer Science and Engineering

Laboratory Manual

Programming with C++ Lab

(18CS301)

NMAM INSTITUTE OF TECHNOLOGY

(An Autonomous Institution affiliated to VTU, Belgaum)

(NBA Accredited, ISO 9001:2008 Certified)

Nitte – 574110, Karkala, Udupi District, Karnataka, India

Course Title: ***Programming with C++ Lab***	*Course Code*: **CS305**
Duration of SEE: **3 hrs**	**No. of Hours/Week: 02**
SEE Marks: **50**	*CIE Marks*: **50**
Lab Manual Author: **Mr. Sannidhan M S**	*Year: 2019-20*

Prerequisites:

- This subject requires the student to know C programming language

Marks Distribution

MSE: 50 Marks

Continuous Evaluation	: 30 Marks
Test	: 20 Marks

SEE:

	PROGRAM 1	PROGRAM 2
WRITEUP	**5**	**5**
EXECUTION	**15**	**15**
VIVA	**10**	
TOTAL	**50**	

List of Programs

SI.No	**Title of the Experiment**
1.	Write C++ program to design a class called BankAccount. Include following data members like name of the depositor, account number and balance. Use following member functions a) to initialize values b) deposit an amount c) to withdraw an amount d) to display name and balance.
2.	Write a C++ program to create a class called COMPLEX and implement the following overloading function ADD that return a COMPLEX number. i. ADD(a,c2);- where a is an integer(real part) & c2 is a complex no. ii. ADD(c1,c2);- where c1 & c2 are complex nos. use Function overloading(ADD) and Friend function concept for the implementation
3.	Write a C++ program with class Time with data members that represents hours and minutes.Include appropriate member functions to compute time in hours and minutes . (Use of objects as arugemnts).
4.	Given that an EMPLOYEE class ontains following members. Data members:Eno, Ename and salary Member functions: to read the data , to print data members.Write a C++ program to read the data of N employees and display detials of each employee**.(use Array of objects concept).**
5.	Write a C++ program with two classes A and B with one integer data member in each class. Write member functions to read and display, place a **friend function** in these classes which takes the data members of these classes and computes maximum of two data members. Demonstrate using main function.
6.	Write a C++ program to demonstrate to uses of **constructors in derived class concept.** (Any inheritance you can use but constructors in base class should have at least one parameter.)
7.	Write a C++ program to create a class sample with integer ,character and float data members. Demonstrate Constructor Overloading on this class with all types of constructors including default argument constructor.
8.	Write a C++ program to demonstrate the working of dynamic constructors using a class STRNG with string as datamember inside the class, include appropriate member functions to display the object data and a member function to concatenate two strings.
9.	Write a C++ program for the diagram using Hierarchical inheritance. Use your own data members and member functions to display student details.\
10.	Write a C++ program for the diagram using Hybrid inheritance. Use yourown data members and member functions to display student details.
11.	Write a C++ program for the diagram using Virtual Base class concept. Use your own data members and member functions to display student details.Create an array of n objects of result class and demonstrate.
12.	Create a base class shape to store two double type values. Derive two specific classes called triangle and rectangle from the base shape. Add to the base class, a member function get_data() to initalize base class data members and another function display_area() to compute and display the area of figures. Make display_area() as a virtual function and redefine this function in the derived classes to suit the requirements.(Area of rectangle=x*y, Area of triangle=1/2*x*y).
13.	Write a C++ program to overload binary + and – operator to add and subtract two complex numbers. Define relevant data members and member functions for reading and displaying the complex objects
14.	Write a C++ program to apply bubble sort on an array of integer and float use the concept of function template
15.	Write a C++ program to implement queue operations for integer and float data types. Use concept of class template

1. Write C++ program to design a class called BankAccount. Include following data members like name of the depositor, account number and balance. Use following member functions a) to initialize values b) deposit an amount c) to withdraw an amount d) to display name and balance.

```
#include <iostream>

using namespace std;
class BankAccount
{
    char name[20];
    int accno;
    int balance;
public:
    void read()
    {
        cout<<"Enter name"<<endl;
        cin>>name;
        cout<<"Enter Accno"<<endl;
        cin>>accno;
        cout<<"Enter Balance"<<endl;
        cin>>balance;
    }
    void print()
    {
        cout<<"Name: "<<name<<endl;
        cout<<"Account Number: "<<accno<<endl;
        cout<<"Balance: "<<balance<<endl;
    }
    void deposit()
    {
        int amount;
        cout<<"Enter amount"<<endl;
        cin>>amount;
        balance=balance+amount;
    }
    void withdraw()
    {
        int amount;
        cout<<"Enter amount to withdraw"<<endl;
        cin>>amount;
        if(amount>balance)
        {
            cout<<"Insufficient"<<endl;
        }
        else
        {
            balance=balance-amount;
        }
    }
    void put_balance()
    {
       cout<<"Account balance is: "<<balance<<endl;
    }

};
int main()
{
   BankAccount b1;
   b1.read();
   b1.print();
   b1.deposit();
   b1.withdraw();
   b1.put_balance();
    return 0;
}
```

2. Write a C++ program to create a class called COMPLEX and implement the following overloading function ADD that return a COMPLEX number.
i. ADD(a,c2);- where a is an integer(real part) & c2 is a complex no.
ii. ADD(c1,c2);- where c1 & c2 are complex nos.
use Function overloading(ADD) and Friend function concept for the implementation

```
#include <iostream>

using namespace std;

class COMPLEX
{
private:int r, i;
public: void read();
                void disp();
                friend COMPLEX ADD(int x, COMPLEX c);
                friend COMPLEX ADD(COMPLEX c1,COMPLEX c2);
};
void COMPLEX::read()
{
        cin>>r>>i;
}
void COMPLEX::disp()
{
        cout<<r<<"+"<<i<<"i";
}
COMPLEX ADD(int x, COMPLEX c)
{
        COMPLEX t;
        t.r=x+c.r;
        t.i=c.i;
        return t;
}
```

```
COMPLEX ADD(COMPLEX c1,COMPLEX c2)
{
        COMPLEX t;
        t.r=c1.r+c2.r;
        t.i=c1.i+c2.i;
        return t;
}
int main()
{
        COMPLEX c1,c2,c3,c4;
        int a;
        cout<<"Enter 1st Complex no :";
        c1.read();
        cout<<"Enter 2nd Complex no :";
        c2.read();
        cout<<"Enter the value of a:";
        cin>>a;
        c3=ADD(a,c2);
        c4=ADD(c1,c2);
        cout<<"\n1st Complex no :";
        c1.disp();
        cout<<"\n2nd Complex no :";
        c2.disp();
        cout<<"\n\nADD(a,c2) :";
        c3.disp();
        cout<<"\nADD(c1,c2) :";
        c4.disp();
        return 0;
}
```

3**. Write a C++ program with class Time with data members that represents hours and minutes.Include appropriate member functions to compute time in hours and minutes . (Use of objects as arugemnts).**

```
#include <iostream>

using namespace std;
class Time
{
   int hours;
   int minutes;
public:
   void read(int h,int m)
   {
      hours=h;
      minutes=m;
   }
   void put()
   {
        cout<<"hours: "<<hours<<endl;
        cout<<"minutes: "<<minutes<<endl;
    }
    void compute(Time t1,Time t2)
    {
        int h=(t1.minutes+t2.minutes)/60;
        minutes=(t1.minutes+t2.minutes)%60;
        hours=t1.hours+t2.hours+h;
    }
};
int main()
{
 Time A,B,C;
 A.read(4,40);
 B.read(4,40);
 cout<<"A: "<<endl;
 A.put();
 cout<<"B: "<<endl;
 B.put();
 C.compute(A,B);
 cout<<"C: "<<endl;
 C.put();
    return 0;
}
```

4. **Given that an EMPLOYEE class ontains following members. Data members:Eno, Ename and salary Member functions: to read the data , to print data members.Write a C++ program to read the data of N employees and display detials of each employee.(use Array of objects concept).**

```
#include <iostream>

using namespace std;
class Employee
{

    char ename[20];
    int eno;
    int esalary;
public:
    void read()
    {
        cout<<"Enter name"<<endl;
        cin>>ename;
        cout<<"Enter enumber"<<endl;
        cin>>eno;
        cout<<"Enter esalary"<<endl;
        cin>>esalary;
    }
    void print()
    {
        cout<<"Name: "<<ename<<endl;
        cout<<"Enumber: "<<eno<<endl;
        cout<<"Esalary: "<<esalary<<endl;
    }

};
int main()
{
   Employee e[10];
   int n;
   cout<<"Enter number of employees"<<endl;
   cin>>n;
   for(int i=0;i<n;i++)
   {
       cout<<"Eneter "<<i+1<<"Employee details"<<endl;
       e[i].read();

   }
   for(int i=0;i<n;i++)
   {
       cout<<i+1<<"Employee details"<<endl;
       e[i].print();

   }
    return 0;
}
```

5. **Write a C++ program with two classes A and B with one integer data member in each class. Write member functions to read and display, place a friend function in these classes which takes the data members of these classes and computes maximum of two data members. Demonstrate using main function.**

```
#include <iostream>

using namespace std;
class B;
class A
{
    int n;
public:
    void read(int a)
    {
        n=a;
    }
    void display()
    {
        cout<<"value= "<<n<<endl;
    }
    friend void maximum(A obj1,B obj2);
};
class B
{
    int n;
public:
    void read(int a)
    {
        n=a;
    }
    void display()
    {
        cout<<"value= "<<n<<endl;
    }
    friend void maximum(A obj1,B obj2);
};
void maximum(A obj1,B obj2)
{
    if(obj1.n>obj2.n)
    {
        cout<<"maximum value is: "<<obj1.n<<endl;
    }
    else
    {
        cout<<"maximum value is: "<<obj2.n<<endl;
    }
}
int main()
{
   A a1;
   a1.read(20);
   B b1;
   b1.read(10);
   cout<<"A: "<<endl;
   a1.display();
   cout<<"B: "<<endl;
   b1.display();
   maximum(a1,b1);
    return 0;
}
```

6. **Write a C++ program to demonstrate to uses of constructors in derived class concept. (Any inheritance you can use but constructors in base class should have at least one parameter.)**

```
#include <iostream>

using namespace std;
class Alpha
{
protected:
   int n1;
public:
   Alpha(int x)
   {
      cout<<"Alpha constructed"<<endl;
    n1=x;
   }
   void putAlpha()
   {
      cout<<"n1: "<<n1<<endl;
   }
};
class Beta
{
protected:
   int n2;
public:
   Beta(int x)

      {
         cout<<"Beta Constructed"<<endl;
         n2=x;
      }
      void putBeta()
   {
      cout<<"n2: "<<n2<<endl;
   }
};
class Gama:public Alpha,public Beta
{
   int n3;
public:
   Gama(int x,int y,int z):
      Alpha(x),
      Beta(y)
      {
         cout<<"Gama Constructed"<<endl;
         n3=z;
      }
      void putGama()
    {
        cout<<"n3: "<<n3<<endl;
    }
};
int main()
{
  Gama g1(10,20,30);
   g1.putAlpha();
   g1.putBeta();
   g1.putGama();
    return 0;
}
```

7. **Write a C++ program to create a class sample with integer ,character and float data members. Demonstrate Constructor Overloading on this class with all types of constructors including default argument constructor.**

```
#include <iostream>

using namespace std;
class Sample
{
    int i;
    char c;
    double d;
public:
    Sample()
    {
        i=0;
        c='\0';
        d=0.0;
    }
    Sample(int x,char y,double z)
    {
         i=x;
         c=y;
         d=z;
    }
    Sample(Sample &s)
    {
         i=s.i;
         c=s.c;
         d=s.d;
    }
Sample(int x,double z,char y='s')
{
    i=x;
    c=y;
    d=z;
}
   void display()
   {
      cout<<"i= "<<i<<endl;
      cout<<"c= "<<c<<endl;
      cout<<"f= "<<d<<endl;
   }
};

int main()
{
   Sample s1;
   cout<<"S1: "<<endl;
   s1.display();
   Sample s2(10,'a',5.6);
   cout<<"S2: "<<endl;
   s2.display();
   Sample s3(30,4.54);
   cout<<"S3: "<<endl;
   s3.display();
   Sample s4(s2);
   cout<<"S4: "<<endl;
   s4.display();
   return 0;
}
```

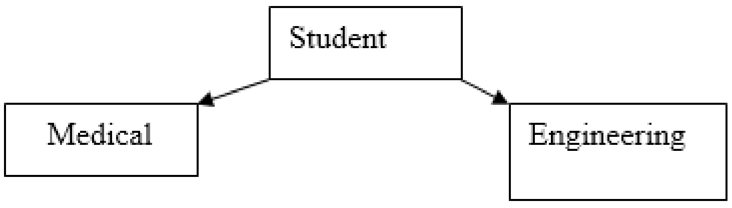
8. Write a C++ program to demonstrate the working of dynamic constructors using a class STRNG with string as datamember inside the class, include appropriate member functions to display the object data and a member function to concatenate two strings.

```
#include <iostream>
#include<string.h>
using namespace std;
class String
{
   int length;
   char *name;
public:
   String(){}
   String(char s[])
   {
      length=strlen(s);
      name=new char[length+1];
      strcpy(name,s);
   }
   void join(String A,String B)
   {
      length=A.length+B.length;
      name=new char[length+1];
      name=strcpy(name,A.name);
```

```
        name=strcat(name,B.name);
    }
    void display()
    {
        cout<<"name= "<<name<<endl;
    }
};
int main()
{
   String s1("wel");
   String s2("fare");
   cout<<"s1: "<<endl;
   s1.display();
   cout<<"s2: "<<endl;
   s2.display();

   String s3;
   s3.join(s1,s2);
   cout<<"s3: "<<endl;
   s3.display();
   String s4("come");
   cout<<"s4: "<<endl;
   s4.display();
   String s5;
   s5.join(s1,s4);
   cout<<"s5: "<<endl;
   s5.display();
   String s6("done");
   cout<<"s6: "<<endl;
   s6.display();
   String s7;
   s7.join(s1,s6);
   cout<<"s7: "<<endl;
   s7.display();
    return 0;
}
```

9. **Write a C++ program for the diagram using Hierarchical inheritance. Use your own data members and member functions to display student details.**

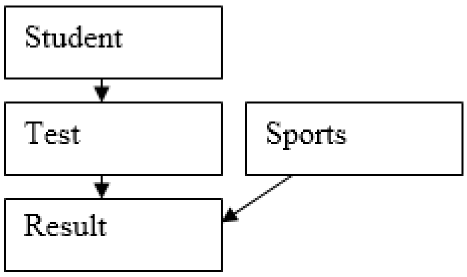


```
#include <iostream>
using namespace std;
class student
{
protected:
char name[30];
int age;
intusn;
public:
voidgetstudent()
    {
cout<<"Enter name "<<endl;
cin>>name;
cout<<"Enter age "<<endl;
cin>>age;
cout<<"Enter usn "<<endl;
cin>>usn;
    }
};
classmedical:public student
{
int year;
public:
voidgetmedical()
    {
cout<<"Enter year"<<endl;
cin>>year;
    }
void display()
    {
cout<<"Medical student details: "<<endl;
cout<<"Name: "<<name<<endl;
cout<<"Age: "<<age<<endl;
cout<<"USN: "<<usn<<endl;
cout<<"Year: "<<year<<endl;
    }
};
classengineering:public student
{
intsem;
char branch[30];
public:
voidgetengineering()
    {
cout<<"Enter sem"<<endl;
cin>>sem;
cout<<"Enter branch"<<endl;
cin>>branch;
    }
void display()
```

```
   {
cout<<"Engineering student details: "<<endl;
cout<<"Name: "<<name<<endl;
cout<<"Age: "<<age<<endl;
cout<<"USN: "<<usn<<endl;
cout<<"Semester: "<<sem<<endl;
cout<<"Branch: "<<branch<<endl;
   }

};
int main()
{
medical m1;
cout<<"Enter medical student details"<<endl;
m1.getstudent();
m1.getmedical();
m1.display();
engineering e1;
cout<<"Enter engineering student details"<<endl;
e1.getstudent();
e1.getengineering();
e1.display();
return 0;
}
```

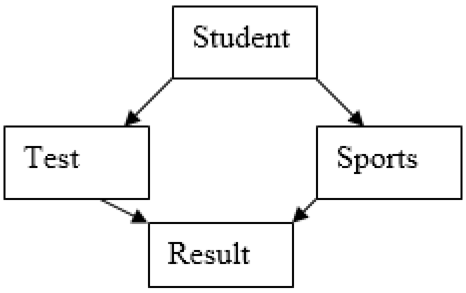
10. Write a C++ program for the diagram using Hybrid inheritance. Use yourown data members and member functions to display student details.



```
#include <iostream>
#include<string.h>
using namespace std;
class Student
{
protected:
   char name[20];
   int usn;
public:
   void read(char n[],int u)
   {
      strcpy(name,n);
      usn=u;
```

```
    }
    void print()
    {
        cout<<"name: "<<name<<endl;
        cout<<"usn: "<<usn<<endl;
    }
};
class Test:public Student
{
protected:
    int sub1;
    int sub2;
public:
    void getmarks(int s1,int s2)
    {
        sub1=s1;
        sub2=s2;
    }
    void putmarks()
    {
        cout<<"sub1: "<<sub1<<endl;
        cout<<"sub2: "<<sub2<<endl;
    }
};
class Sports
{
protected:
    int score;
public:
    void getscore(int s)
    {
        score=s;
    }
    void putscore()
    {
        cout<<"score= "<<score<<endl;
    }
};
class Result:public Test,public Sports
{
    int total;
public:
    void display()
    {
        total=sub1+sub2+score;
        print();
        putmarks();
        putscore();
        cout<<"total: "<<total<<endl;
    }
};
int main()
{
    Result r1;
    r1.read("xyz",20);
    r1.getmarks(25,35);
    r1.getscore(3);
    r1.display();
    return 0;
}
```

11. **Write a C++ program for the diagram using Virtual Base class concept. Use your own data members and member functions to display student details.Create an array of n objects of result class and demonstrate.**



```
#include <iostream>
#include<string.h>
using namespace std;
class Student
{
protected:
    char name[20];
    int usn;
public:
    void read()
    {
        cout<<"Enter name"<<endl;
        cin>>name;
        cout<<"Enter usn"<<endl;
        cin>>usn;
    }
    void print()
    {
        cout<<"name: "<<name<<endl;
        cout<<"usn: "<<usn<<endl;
    }
};
class Test:virtual public Student
{
```

```
protected:
    int sub1;
    int sub2;
public:
    void getmarks()
    {
        cout<<"Enter sub1 marks"<<endl;
        cin>>sub1;
        cout<<"Enter sub2 marks"<<endl;
        cin>>sub2;

    }
    void putmarks()
    {
        cout<<"sub1: "<<sub1<<endl;
        cout<<"sub2: "<<sub2<<endl;
    }
};
class Sports :virtual public Student
{
protected:
    int score;
public:
    void getscore()
    {
        cout<<"Enter score"<<endl;
        cin>>score;
    }
    void putscore()
    {
        cout<<"score= "<<score<<endl;
    }
};
class Result:public Test,public Sports
{
    int total;
public:
    void display()
    {
        total=sub1+sub2+score;
        print();
        putmarks();
        putscore();
        cout<<"total: "<<total<<endl;

    }
};
int main()
{
    Result r[10];
    int n;
    cout<<"Enter number of students: "<<endl;
    cin>>n;
    for(int i=0;i<n;i++)
    {
        r[i].read();
        r[i].getmarks();
        r[i].getscore();
    }
   for(int i=0;i<n;i++)
    {
        r[i].display();
    }
    return 0;
}
```

12. **Create a base class shape to store two double type values. Derive two specific classes called triangle and rectangle from the base shape. Add to the base class, a member function get_data() to initalize base class data members and another function display_area() to compute and display the area of figures. Make display_area() as a virtual function and redefine this function in the derived classes to suit the requirements.(Area of rectangle=x*y, Area of triangle=1/2*x*y).**

```
#include <iostream>

using namespace std;
class shape
{
    public:
    double x;
    double y;
    void get_data();
    virtual void display_area()=0;
};
void shape::get_data()
{
    cout<<"Enter the values: ";
    cin>>x>>y;
}

class triangle:public shape
{
    public:
    void display_area();
};
void triangle::display_area()
{
    cout<<"The area of the triangle is: "<<"\n";
    cout<<0.5*x*y<<"\n";
}
```

```
class rectangle:public shape
{
     public:
     void display_area();
};
void rectangle::display_area()
{
    cout<<"The area of the rectangle is: "<<"\n";
    cout<<x*y<<"\n";
}
int main()
{
    int ch;
    char c;
    triangle tr;
    rectangle re;
    shape *sptr;

    do
    {
        ch=0;
        cout<<"Enter 1 r area of triangle, 2 for area of rectangle: ";
        cin>>ch;
        switch(ch)
        {

            case 1:
            sptr=&tr;
            sptr->get_data();
            sptr->display_area();
            break;
            case 2:
            sptr=&re;
            sptr->get_data();
            sptr->display_area();
            break;
            default:
            cout<<"Choice mismatch"<<"\n";
            break;
        }
        cout<<"Do you want to continue: ";
        cin>>c;
    }while(c=='Y'||c=='y');
    return 0;
}
```

13. **Write a C++ program to overload binary + and – operator to add and subtract two complex numbers. Define relevant data members and member functions for reading and displaying the complex objects**

```
#include <iostream>

using namespace std;
class Complex
{
    int real;
    float imag;
public:
    void read(int r,float i)
    {
        real=r;
        imag=i;
    }
    void display()
    {
        cout<<real<<"+i"<<imag<<endl;
    }
    Complex operator+(Complex c)
    {
        Complex temp;
        temp.real=real+c.real;
        temp.imag=imag+c.imag;
        return temp;
    }
     Complex operator-(Complex c)
    {
        Complex temp;
        temp.real=real-c.real;
        temp.imag=imag-c.imag;
        return temp;
    }
};
int main()
{
   Complex c1,c2,c3,c4;
   c1.read(1,1.4);
   c2.read(3,2.2);
   cout<<"C1: "<<endl;
   c1.display();
   cout<<"C2: "<<endl;
   c2.display();
   c3=c1+c2;
   cout<<"C3:"<<endl;
   c3.display();
   c4=c2-c1;
   cout<<"C4: "<<endl;
```

```
  c4.display();
   return 0;
}
```

14. **Write a C++ program to apply bubble sort on an array of integer and float use the concept of function template**

```
#include <iostream>

using namespace std;
template<class T>
void bsort(T arr[],int n)
{
    for(int i=0;i<n-1;i++)
    {
        for(int j=0;j<n-1-i;j++)
        {
            if(arr[j]>arr[j+1])
            {
                T temp=arr[j];
                arr[j]=arr[j+1];
                arr[j+1]=temp;
            }
        }
    }
}
int main()
{
    cout<<"Integer array sort"<<endl;
    int ia[30],n;
    cout<<"Enter the number of elements"<<endl;
    cin>>n;
    cout<<"Enter array elements"<<endl;
    for(int i=0;i<n;i++)
    {
        cin>>ia[i];
    }
    cout<<"Array elements before sorting"<<endl;
    for(int i=0;i<n;i++)
    {
        cout<<ia[i]<<"\t";
    }
    cout<<endl;
    bsort(ia,n);
    cout<<"Array elements after sorting"<<endl;
    for(int i=0;i<n;i++)
    {
        cout<<ia[i]<<"\t";
    }
    cout<<endl;
cout<<"Floating array sort"<<endl;
    float fa[30];
    cout<<"Enter the number of elements"<<endl;
    cin>>n;
    cout<<"Enter array elements"<<endl;
    for(int i=0;i<n;i++)
    {
        cin>>fa[i];
    }
    cout<<"Array elements before sorting"<<endl;
    for(int i=0;i<n;i++)
    {
        cout<<fa[i]<<"\t";
    }
    cout<<endl;
    bsort(fa,n);
    cout<<"Array elements after sorting"<<endl;
    for(int i=0;i<n;i++)
    {
        cout<<fa[i]<<"\t";
    }
    cout<<endl;

    return 0;
}
```

15**. Write a C++ program to implement queue operations for integer and float data types. Use concept of class template**

```
#include <iostream>
#define max 5
using namespace std;

template<class T>
class Queue
{
    int f=0;
    int r=-1;
    T arr[max];
public:

    void insertq()
    {
        if(r==max-1)
        {
            cout<<"queue full"<<endl;
        }
        else
        {
            T ele;
            cout<<"Enter element to be inserted"<<endl;
            cin>>ele;
            r=r+1;
            arr[r]=ele;
        }
    }
     void deleteq()
    {
        if(f>r)
        {
            cout<<"queue empty, cannot delete"<<endl;
        }
        else
        {
           T ele=arr[f];
            f=f+1;
            cout<<"Element deleted: "<<ele<<endl;
        }
    }
    void display()
    {
        if(f>r)
        {
            cout<<"queue empty"<<endl;
        }
        else
        {
            cout<<"Elements: "<<endl;
            for(int i=f;i<=r;i++)
            {
                cout<<arr[i]<<"\t";
            }
            cout<<endl;
        }
    }
    void operation()
    {
        int choice;
        for(;;)
        {
            cout<<"Enter 1: insert\n2:delete\n3:Display\n4:Return"<<endl;
            cin>>choice;
            switch(choice)
            {
                case 1:insertq();
                break;
                case 2:deleteq();
                break;
                case 3:display();
                break;
```

```
                default:return;
            }
        }
    }
};
int main()
{
    Queue<int>q1;
    Queue<float>q2;
    int choice;
    for(;;)
    {
        cout<<"1:Integer Operation\n2:Floating Point\n3:Exit"<<endl;
        cin>>choice;
        switch(choice)
        {
            case 1:q1.operation();
            break;
            case 2:q2.operation();
            break;
            default: return(0);
        }
    }



    return 0;
}
```