FirstRanker.com
Firstranker's choice
www.FirstRanker.com www.FirstRanker.com

OBJECT ORIENTED PROGRAMMING THROUGH C++

I B.TECH II SEMESTER

(R16- Regulation)

Academic Year: 2018-2019

(LAB MANUAL)

www.FirstRa

Prepared By

Mrs. M R S Lavanya Devi, M.Tech.

ASSISTANT PROFESSOR

DEPARTMENT OF COMPUTER SCIENCE ENGINEERING

VSM
COLLEGE OF ENGINEERING
RAMACHANDRAPURAM, E.G.Dist
AN ISO 9001 : 2008 CERTIFIED INSTITUTION
Affiliated to JNTUK, KAKINADA

www.FirstRanker.com

FirstRanker.com
Firstranker's choice
www.FirstRanker.com www.FirstRanker.com

OBJECT-ORIENTED PROGRAMMING LAB

OBJECTIVE

- To strengthen their problem solving ability by applying the characteristics of an object- oriented approach.
- To introduce object oriented concepts in C++ and Java.

Programmig:

Exercise – 1 (Basics)

Write a Simple Program on printing "Hello World" and "Hello Name" where name is the input from the user

- Convert any two programs that are written in C into C++
- Write a description of using g++ (150 Words)

Exercise – 2 (Expressions Control Flow)

Write a Program that computes the simple interest and compound interest payable on principalamount(inRs.)ofloanborrowedbythecustomerfromabankforagiverperiodof time (in years) at specific rate of interest. Further determine whether the bank will benefit by charging simple interest or compound interest.

- WriteaProgramtocalculatethefareforthepassengerstravelinginabus. WhenaPassenger enters the bus, the conductor asks "What distance will you travel?" On knowing distance from passenger (as an approximate integer), the conductor mentions the fare to the passenger according to following criteria.

www.FirstRanker.com

Exercise – 3 (Variables, Scope, Allocation)

- Write a program to implement call by value and call by reference using reference variable.
- Write a program to illustrate scope resolution, new and delete Operators. (Dyanamic Memory Allocation)
- Write a program to illustrate Storage classes

www.FirstRanker.com

FirstRanker.com
Firstranker's choice
www.FirstRanker.com www.FirstRanker.com

- Write a program to illustrate Enumerations

Exercises –4 (Functions)

Write a program illustrating Inline Functions

- Write a program illustrate function overloading. Write 2 overloading functions for power.
- Write a program illustrate the use of default arguments for simple interest function.

Exercise -5 (Functions –Exercise Continued)

- Write a program to illustrate function overloading. Write 2 overloading functions for adding two numbers
- Write a program illustrate function template for power of a number.
- Write a program to illustrate function template for swapping of two numbers.

Exercise -6 (Classes Objects) Create a Distance class with:

- feet and inches as data members
- member function to input distance
- member function to output distance
- member function to add two distance objects

- Write a main function to create objects of DISTANCE class. Input two distances and output the sum.
- Write a C++ Program to illustrate the use of Constructors and Destructors (use the above program.)

c) Write a program for illustrating function overloading in adding the distance between objects (use the above problem)

- Write a C++ program demonstrating a BankAccount with necessary methods and variables

Exercise – 7 (Access)

Write a program for illustratingAccess Specifiers public, private, protected

- Write a program implementing Friend Function
- Write a program to illustrate this pointer
- Write a Program to illustrate

pointer to a class d)

Exercise -8 (Operator Overloading)

a). Write a program to Overload Unary, and Binary Operators as Member Function, and Non Member Function.

- Unary operator as member function
- Binary operator as nonmember function

www.FirstRanker.com

FirstRanker.com
Firstranker's choice
www.FirstRanker.com www.FirstRanker.com

b). Write a c ++ program to implement the overloading assignment = operator

c).Write a case study on Overloading Operators and Overloading Functions (150 Words)

Exercise -9 (Inheritance)

- Write C++ Programs and incorporating various forms of Inheritance
  - Single Inheritance
  - Hierarchical Inheritance
  - Multiple Inheritances
  - Multi-level inheritance
  - Hybrid inheritance
- Write a program to show Virtual Base Class
- Write a case study on using virtual classes (150 Words)

Exercise-10 (Inheritance –Continued)

- Write a Program in C++ to illustrate the order of execution of constructors and destructors in inheritance
- Write a Program to *show* how *constructors* are invoked in *derived class*

Exercise -11 (Polymorphism)

- Write a program to illustrate runtime polymorphism
- Write a program to illustrate this pointer
- Write a program illustrates pure virtual function and calculate the area of different shapes by using abstract class.
- Write a case study on virtual functions (150 Words)

Exercise -12(Templates)

- Write a C++ Program to illustrate template class
- Write a Program to illustrate class templates with multiple parameters
- Write a Program to illustrate member function templates

Exercise -13 (Exception Handling)

a).Write a Program for Exception Handling

Divide by zero b). Write a Program to

rethrow an Exception

Exercise -14 (STL)

- Write a Program to implement List and List Operations

www.FirstRanker.com

www.FirstRanker.com

FirstRanker.com
Firstranker's choice
www.FirstRanker.com www.FirstRanker.com

- Write a Program to implementVector andVector Operations

Exercise -15 (STLContinued)

- Write a Program to implement Deque and Deque Operations
- Write a Program to implement Map and Map Operations

OUTCOMES:

- Explain whatconstitutesanobject-oriented approachtoprogrammingandidentify potential benefits of object-oriented programming over other approaches.
- Apply an object-oriented approach to developing applications of varying complexities

www.FirstRanker.com

www.FirstRanker.com

FirstRanker.com
Firstranker's choice
www.FirstRanker.com www.FirstRanker.com

Exercise – 1 (Basics)

Write a Simple Program on printing "Hello World" and "Hello Name" where name is the input from the user

- **Convert any two programs that are written in C into C++**
- **Write a description of using g++ (150 Words)**

1a) Aim: Write a Simple Program on printing "Hello World" and "Hello Name" where name is the input from the user.

Code:

```
#include<iostream>

#include<iomanip>

using namespace std;

main()

{

char name[15];

cout<<"Enter any name:";

cin>>name;

cout<<"Hello World"<<setw(10)<<"Hello "<<name;

}
```

www.FirstRanker.com

Output:

```
Enter any name: ram

Hello World    Hello ram
```


www.FirstRanker.com

FirstRanker.com
Firstranker's choice
www.FirstRanker.com www.FirstRanker.com

Aim: C++ Program to illustrate Reverse of the given no.

```
#include<iostream>
using namespace std;
int main()
{
int num,r,rev=0;
cout<<"Enter any no:";
cin>>num;
while(num!=0)
{
r=num%10;
rev=rev*10+r;
num=num/10;
}
cout<<"Reverse Number:"<<rev;
return 0;
}
```

www.FirstRanker.com

Output:

Enter any no:152

Reverse Number:251

www.FirstRanker.com

FirstRanker.com
Firstranker's choice
www.FirstRanker.com www.FirstRanker.com

Aim : C++ program for Armstrong or not.

Code:

```
#include<iostream>
using namespace std;
int main()
{
  int origNum, num, r, sum = 0;
    cout << "Enter a positive  integer: ";
  cin >> origNum;

  num = origNum;


  while(num != 0)

  {

     r= num % 10;

      sum += r*r*r;

      num /= 10;

  }


  if(sum == origNum)

    cout << origNum << " is an Armstrong number.";

  else

    cout << origNum << " is not an Armstrong number.";

  return 0;

}
```

Output:

./a.out

Enter a positive integer: 153
153 is an Armstrong number
1b) Aim: Write a description of using g++ (150 words)

www.FirstRanker.com

www.FirstRanker.com

FirstRanker.com
Firstranker's choice
www.FirstRanker.com www.FirstRanker.com

Write a description of using g++ (150 Words)
Description:

GCC stands for " GNU Compiler Collection ". As its name states, GCC is a collection of compilers for several programming languages, and C++ is just one of them.

g++ is known as a **compiler**, a program that will take your C++ source code and compile it into a binary file that can be executed to actually run your program.the official compiler of the GNU operating system, GCC has been adopted as the standard compiler by many other modern Unix-like computer operating systems, including Linux and the BSD family.

- **Installation of g++ in Ubuntu :**
  - Do the following on a terminal:

    sudo apt-get update

    to update your package list with the most recent version of g++ enter you password and press Return and then,
  - Do:

    sudo apt-get install g++

    to install g++.

Compile a Source Code File by g++ :

To compile your source code, you must first reside in the directory that your source code files are available.

g++ filename.cpp

This is the most basic way to compile your source code. After you type this command, 1 of 2 things will happen. Either your source code will be without error and the exectuable will be written, returning you back to the command prompt OR your code will have a few errors in it and a display of the errors.

If no errors are found in the program,the above command will produce an executable in the same directory called a.out which you can run by typing this in your terminal:

./a.out

www.FirstRanker.com

www.FirstRanker.com

FirstRanker.com
Firstranker's choice
www.FirstRanker.com www.FirstRanker.com

Exercise – 2 (Expressions Control Flow)

Write a Program that computes the simple interest and compound interest payable on principalamount(inRs.) of loan borrowed by the customer from a bank for a given period of time (in years) at specific rate of interest. Further determine whether the bank will benefit by charging simple interest or compound interest.

```
#include<iostream>
using namespace std;
int main()
{
int p,t,r,Si,Ci;
cout<<"Enter Principal, Time and Rate : ";
cin>>p>>t>>r;
Si=(p*t*r)/100;
Ci=p*((1+r/100)^t)-p;
cout<<"Simple Interest is:"<<Si<<"\n";
cout<<"Compound Interest is :"<<Ci<<"\n";
if(Si>Ci)
cout<<"bank will benefit by charging Simple Interest \n";
else
cout<<"bank will benefit by charging Compound Interest ";
return 0;
}
```

Output:

```
vsm@cselab1sys23:~/Desktop$ g++ first.cpp
vsm@cselab1sys23:~/Desktop$ ./a.out
Enter Principal, Time and Rate : 10000
5
12
Simple Interest is:6000
Compound Interest is :30000
bank will benefit by charging Compound Interest
```

www.FirstRanker.com

2b) Aim: Write a Program to calculate the fare for the passengers travelingin a bus. When a Passenger enters the
bus, the conductor asks "What distance will you travel?" On knowing distance from passenger
(as an approximate integer), the conductor mentions the fare to the passenger according to following criteria.
If minimum distance 2 km—fare 6 Rs/-
if distance less than 14 kms – fare 1Rs/- for 1 km
if distance grater than 14 kms- fare 2Rs/- for 1 km

www.FirstRanker.com

FirstRanker.com
Firstranker's choice
www.FirstRanker.com www.FirstRanker.com


```
#include<iostream>

using namespace std;

int main()

{

float distance,ticket_fare;

cout<<"how much distance you want to travel?";

cin>>distance;

if(distance<=2)

ticket_fare=6;

else if(2<distance<=14)

ticket_fare=6+((int)distance-2)*1;

else

ticket_fare=18+((int)distance-16);

cout<<"the total fare is :"<<ticket_fare;

return 0;

}
```

Output:

```
./a.out
vsm@cselab1sys23:~/Desktop$ g++ first.cpp

vsm@cselab1sys23:~/Desktop$ ./a.out

how much distance you want to travel? 16

the total fare is : 20
vsm@cselab1sys23:~/Desktop$ ./a.out
```

www.FirstRanker.com

www.FirstRanker.com

FirstRanker.com
Firstranker's choice
www.FirstRanker.com www.FirstRanker.com

Exercise – 3 (Variables, Scope, Allocation)
Write a C++ Program to illustrate call by Value..

Code:

```
#include<iostream>

using namespace std;

void swap(int a, int b)

{

   int temp = 0;

   temp = a;

   a = b;

   b = temp;

  cout<< "After swap in sub";

  cout<< "\na = "<< a <<"\n"<<"b = "<<b<<"\n";

}
int main(void)

{

     int a = 10;

     int b = 20;

  cout<< "\na = "<< a <<"\n"<<"b = "<<b<<"\n";

  swap(a,b);

  cout<< "After swap in main";

  cout<< "\na = "<< a <<"\n"<<"b = "<<b<<"\n";

  return 0;

}
```

www.FirstRanker.com

www.FirstRanker.com

FirstRanker.com
Firstranker's choice
www.FirstRanker.com www.FirstRanker.com

Output:

```
vsm@cselab1sys23:~/Desktop$ g++ first.cpp

vsm@cselab1sys23:~/Desktop$ ./a.out
a = 10

b = 20

After swap in sub

a = 20

b = 10

After swap in main

a = 10

b = 20
```

Write a C++ program to illustrate call by Reference using reference variable.

Code:

```
#include<iostream>

using namespace std;
void swap(int *a, int *b)
{

   int temp = 0;

   temp = *a;

   *a = *b;

   *b = temp;

cout<< "After swap in sub";

cout<< "\na = "<< *a <<"\n"<<"b = "<<*b<<"\n";

}
int main(void)
```

www.FirstRanker.com

www.FirstRanker.com

FirstRanker.com
Firstranker's choice
www.FirstRanker.com www.FirstRanker.com


```
{
   int a = 10;
   int b = 20;
   cout<< "Before swap in main";
   cout<<"\na = "<< a <<"\n"<<"b = "<<b<<"\n";
   swap(&a,&b);
   cout<<"After swap in main";
   cout<< "\na = "<< a <<"\n"<<"b = "<<b<<"\n";
   return 0;
}
```

Output:

```
vsm@cselab1sys23:~/Desktop$ g++ first.cpp

vsm@cselab1sys23:~/Desktop$ ./a.out

Before swap in main

a = 10

b = 20

After swap in sub

a = 20

b = 10

After swap in main

a = 20

b = 10
```

www.FirstRanker.com

www.FirstRanker.com

FirstRanker.com
Firstranker's choice
www.FirstRanker.com www.FirstRanker.com

Aim: C++ Program for illustrating scope resolution operator.

Code:

```
#include <iostream>
using namespace std;
 char c = 'a';     // global variable
 int main( )
{
     char c = 'b';   //local variable
     cout << "Local  variable: " << c << "\n";
     cout << "Global variable: " << ::c << "\n";  //using scope resolution operator
     return 0;
}
```

Output:

```
Localvariable: b
Globalvariable: a
```

4. Aim: C++ Program for Illustrating new and delete operators.

Code:

```
#include<iostream>

using namespace std;

int main()

{

int *pa=new int;
int *pb=new int;
int *psum = new int;
cout<<"Enter two integers:";

cin>>*pa>> *pb;

*psum=*pa + *pb;

cout<< "Sum of two integers = "<<*psum;

delete pa;
```

www.FirstRanker.com

www.FirstRanker.com

FirstRanker.com
Firstranker's choice
www.FirstRanker.com www.FirstRanker.com


```
delete pb;

return 0;

}
```

Output:

```
lab1@lab1:~/Desktop$ g++ first.cpp
lab1@lab1:~/Desktop$ ./a.out
Enter two integers:5
6
Sum of two integers = 11
```

5.Aim: C++ Program for illustrating storage classes.

Code:

```
#include <iostream>

int sum(int n1, int n2)
{
 auto int s;        //declaration of auto(local) variable
s = n1+n2;
 return s;
}

int main( )
{
int i = 2, j = 3, k;
k = sum(i, j);
std::cout << "sum is " << k << std::endl;
return 0;
}
```

www.FirstRanker.com

Output:

```
sum is 5
```


www.FirstRanker.com

FirstRanker.com
Firstranker's choice
www.FirstRanker.com www.FirstRanker.com

6.Aim: C++ Program for illustrating enumerations.

Code:

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

enum week { Sunday, Monday, Tuesday, Wednesday, Thursday, Friday, Saturday };
enum seasons { spring = 34, summer = 4, autumn = 9, winter = 32};

int main()
{
   week today;
         seasons s;


   today = Wednesday;
   cout << "Day " << today+1;

s = summer;
   cout << "Summer = " << s << endl;

   return 0;
}
```

output :

```
4
```

www.FirstRanker.com

Exercises –4 (Functions)

Write a program illustrating Inline Functions.

```
#include <iostream>
using namespace std;
inline int cube(int s)
{
   return s*s*s;
}
int main()
{
   cout << "The cube of 3 is: " << cube(3) << "\n";
   return 0;
```


www.FirstRanker.com

FirstRanker.com
Firstranker's choice
www.FirstRanker.com www.FirstRanker.com


```
}
```

Output:

```
The cube of 3 is: 27
```

Write a program illustrate function overloading.

```
#include <iostream>
using namespace std;
class Addition
 {
public:
    int sum(int num1,int num2) {
        return num1+num2;
    }
    int sum(int num1,int num2, int num3) {
       return num1+num2+num3;
    }
  };
int main(void)
 {
    Addition obj;
    cout<<obj.sum(20, 15)<<endl;
    cout<<obj.sum(81, 100, 10);
    return 0;
}
```

Output:

```
35
191
```

Write a program illustrate the use of default arguments for simple interest function.

Code:

```
#include<iostream>

using namespace std;

int main()

{

float amount, principal;
```

www.FirstRanker.com

www.FirstRanker.com

FirstRanker.com
Firstranker's choice
www.FirstRanker.com www.FirstRanker.com


```
int time;
float interest(float p, int n, float r=0.15);
cout<<"Program for simple interest calculation"<<endl;
cout<<"Please enter principal amount:";
cin>>principal;
cout<<"Please enter Time: ";
cin>>time;
amount=interest(principal, time);
cout<<"\nThe amount is:"<<amount;
cout<<"\nPress any key to continue...";
return 0;
}
float interest(float p, int n,float r)
{
return (p+((p*r*n)/100.0));
}
```

Output:

```
lab1@lab1:~/Desktop$ g++ first.cpp
lab1@lab1:~/Desktop$ ./a.out
Program for simple interest calculation
Please enter principal amount:15000
Please enter Time: 5
The amount is:15112.5
```

www.FirstRanker.com

www.FirstRanker.com

FirstRanker.com
Firstranker's choice
www.FirstRanker.com www.FirstRanker.com

Exercise -5 (Functions –Exercise Continued)
Write a program to illustrate function overloading. Write 2 overloading functions for adding two numbers

```
#include <iostream>
using namespace std;
int sum(int, int);
float sum(float, float);
float sum(int, float);
int main()
{
  int num1, num2, x;
  float num3, num4, y;
  cout<<"Enter two integer numbers: ";
  cin>>num1>>num2;
  //This will call the first function
  cout<<"Result: "<<sum(num1, num2)<< endl;
  cout<<"Enter two float numbers: ";
  cin>>num3>>num4;
  //This will call the second function
  cout<<"Result: " <<sum(num3, num4)<< endl;
  cout<<"Enter one int and one float number: ";
  cin>>x>>y;
  //This will call the third function
  cout<<"Result: " <<sum(x, y)<< endl;
  return 0;
}
int sum(int a, int b)
{
  return a+b;
}
float sum(float a, float b)
{
  return a+b;
}
```

www.FirstRanker.com

www.FirstRanker.com

FirstRanker.com
Firstranker's choice
www.FirstRanker.com www.FirstRanker.com


```
float sum(int a, float b){
  return a+b;
}
```

Output:

```
Enter two integer numbers: 21 88
Result: 109
Enter two float numbers: 10.2 30.7
Result: 40.9
Enter one int and one float number: 20 16.4
    Result: 36.4
```

Write a program illustrate function template for power of a number.

```
#include <iostream>
#include <cmath>
using namespace std;
template<class T>
void power(T x, int y)
{
        cout<<pow(x, y)<<endl;
}
int main()
{
        power(2, 4);
        power(2.5, 2);
        return 0;
}
```

www.FirstRanker.com

Output:

```
16
6.25
```

Write a program to illustrate function template for swapping of two numbers.

Code:

```
#include<iostream>

using namespace std;

template <class A> void swapa(A &x,A &y)

{
```


www.FirstRanker.com

FirstRanker.com
Firstranker's choice
www.FirstRanker.com www.FirstRanker.com


```
        A t;

        t=x;x=y;y=t;

        return;

}

int main()

{

        int a=2,b=3;
        cout<<"Before Swapping:"<<endl<<"a="<<a<<endl<<"b="<<b<<endl;
        swapa(a,b);
        cout<<"After Swapping:"<<endl<<"a= "<<a<<"\nb= "<<b<<endl;
        return 0;
}
```

Output:

```
lab1@lab1:~/Desktop$ g++ first.cpp

lab1@lab1:~/Desktop$ ./a.out

Before Swapping:

a=2

b=3

After Swapping:

a= 3

b= 2
```

www.FirstRanker.com

Exercise -6 (Classes Objects) Create a Distance class with:
•feet and inches as data members
•member function to input distance
•member function to output distance
•member function to add two distance objects

Write a main function to create objects of DISTANCE class. Input two distances and output the sum.

www.FirstRanker.com

FirstRanker.com
Firstranker's choice
www.FirstRanker.com www.FirstRanker.com


```
#include<iostream.h>
Class distance
{
  Public:  float d1;
           float d2;
           float sum;
           float Sum(float x, float y)
              {
                  cin>>x;
                  cin>>y;
                 sum = x+y;
                 cout<<"The sum of distances is :"<<sum;

              }
};
int main( )
{
   distance d;
   d.sum( );
}
```

Output:

10
20
The sum of distance is :30

Write a C++ Program to illustrate the use of Constructors and Destructors .

```
#include <iostream>
using namespace std;
class Example{
public:
                //default constructor
                Example(){cout<<"Constructor called."<<endl;}
                //function to print message
                void display(){
                        cout<<"display function called."<<endl;
                }
                //Destructor
                ~Example(){cout<<"Destructor  called."<<endl;}
};
int main(){
        //object creation
        Example objE;

        objE.display();
```

www.FirstRanker.com

www.FirstRanker.com

FirstRanker.com
Firstranker's choice
www.FirstRanker.com www.FirstRanker.com


```
        return 0;
}
```

Output:

```
Constructor called.
display function called.
Destructor called.
```

Write a program for illustrating function overloading in adding the distance between objects

```
#include <iostream>
using namespace std;
class Distance
{
        private:
                int feet;
                int inches;
        public:
                void set_distance()
                {
                        cout<<"Enter feet: ";
                        cin>>feet;
                        cout<<"Enter inches: ";
                        cin>>inches;
                }
                void get_distance()
                {
                        cout<<"Distance is feet= "<<feet<<", inches= "<<inches<<endl;
                }
                void add(Distance d1, Distance d2)
                {
                        feet = d1.feet + d2.feet;
                        inches = d1.inches + d2.inches;
                        feet = feet + (inches / 12);
                        inches = inches % 12;
                }
                void add(Distance *d1, Distance *d2)
                {
                        feet = d1->feet + d2->feet;
                        inches = d1->inches + d2->inches;
                        feet = feet + (inches / 12);
                        inches = inches % 12;
                }
};
```

www.FirstRanker.com

www.FirstRanker.com

FirstRanker.com
Firstranker's choice
www.FirstRanker.com www.FirstRanker.com


```
int main()
{
        Distance d1, d2, d3;
        d1.set_distance();
        d2.set_distance();
        d3.add(d1, d2);
        d3.get_distance();
        d3.add(&d1, &d2);
        d3.get_distance();
        return 0;
}
```

Output :

```
Enter feet: 3
Enter inches: 4
Enter feet: 4
Enter inches: 9
Distance is feet= 8, inches= 1
Distance is feet= 8, inches= 1
```

Write a C++ program demonstrating a BankAccount with necessary methods and variables

```
#include<iostream>

#include<stdlib.h>

using namespace std;

class bank {

        int ac;

        static int reset;

        float balance, amount, shorta;

public:

        void deposit();

        void withdraw();

        void chkbalance();

        int menu();
```

www.FirstRanker.com

www.FirstRanker.com

FirstRanker.com
Firstranker's choice
www.FirstRanker.com www.FirstRanker.com


```
};

int bank::reset=0;

void bank::deposit() {

        cout<<"\nEnter Account Number:";

        cin>>ac;

        if(ac==1001)

        {

                cout<<"\nEnter Amount you want to Deposit:";

                cin>>amount;

                balance=balance+amount;

                cout<<"\n\nMessage: You have successfully deposited Rs."<<amount<<" into
your account.\n\nYour Current Balance is:"<<balance<<"\n\n";

                reset++;

        }

        else

        {

                cout<<"You have entered invalid account number";

        }

}


void bank::withdraw() {

        cout<<"Enter Account Number:";

        cin>>ac;

        if(ac==1001)
```

www.FirstRanker.com

www.FirstRanker.com

FirstRanker.com
Firstranker's choice
www.FirstRanker.com www.FirstRanker.com


```
        {
            cout<<"Enter Amount you want to Withdraw:";
            cin>>amount;
            if(amount>balance)
            {
                shorta=amount-balance;
                cout<<"\n\nYou cannot withdraw Rs."<<amount<<" as your account
balance is Rs."<<balance<<"\n\nYou are short of Rs."<<shorta;
            }
            else
            {
                balance=balance-amount;
                cout<<"You have Withdrawn Rs."<<amount<<" Successfully\n\n Your
account balance is:"<<balance<<"\n\n";
            }
        }
        else
        {
            cout<<"You have entered invalid account number";
        }
}
void bank::chkbalance() {
        cout<<"Enter Account Number:";
        cin>>ac;
        if(reset==0)
```

www.FirstRanker.com

www.FirstRanker.com

FirstRanker.com
Firstranker's choice
www.FirstRanker.com www.FirstRanker.com


```
        {
                balance=0;
        }
        if(ac==1001)
        {
                cout<<"Your Account balance is Rs."<<balance<<"\n\n";
        }
        else
        {
                cout<<"Account Doesn't Exist!!";
        }
}
int main()
{
        bank ob;
        int ch;
        do {
                cout<<"\n\n\n1.Deposit\n2.Withdraw\n3.Balance Enquiry\n4.Exit\n\n";
                ch=ob.menu();
                switch(ch)
                {
                case 1:ob.deposit();
                break;
```

www.FirstRanker.com

www.FirstRanker.com

FirstRanker.com
Firstranker's choice
www.FirstRanker.com www.FirstRanker.com


```
                case 2:ob.withdraw();
                        break;
                case 3:ob.chkbalance();
                        break;
                case 4: exit(1);
                default:cout<<"Invalid Choice!!";
                }
        }while(1);
return 0;
}
int bank::menu()
{
        int ch;
        cout<<"\nEnter your Choice:";
        cin>>ch;
        return ch;
}
```

www.FirstRanker.com

Output:

```
lab1@lab1:~/Desktop$ g++ first.cpp
lab1@lab1:~/Desktop$ ./a.out

1.Deposit
2.Withdraw
```


www.FirstRanker.com

FirstRanker.com
Firstranker's choice
www.FirstRanker.com www.FirstRanker.com

3.Balance Enquiry

4.Exit

Enter your Choice:1

Enter Account Number:524487991

You have entered invalid account number

1.Deposit

2.Withdraw

3.Balance Enquiry

4.Exit

Enter your Choice:1

Enter Account Number:1001

Enter Amount you want to Deposit:500

Message: You have successfully deposited Rs.500 into your account.

Your Current Balance is:500

1.Deposit

2.Withdraw

3.Balance Enquiry

4.Exit

Enter your Choice:2

Enter Account Number:1001

Enter Amount you want to Withdraw:200

You have Withdrawn Rs.200 Successfully

www.FirstRanker.com

www.FirstRanker.com

FirstRanker.com
Firstranker's choice
www.FirstRanker.com www.FirstRanker.com

Your account balance is:300

1.Deposit

2.Withdraw

3.Balance Enquiry

4.Exit

Enter your Choice:3

Enter Account Number:1001

Your Account balance is Rs.300

1.Deposit

2.Withdraw

3.Balance Enquiry

4.Exit

Exercise – 7 (Access Specifiers)

Write a program for illustratingAccess Specifiers public, private, protected.

```
#include<iostream>
using namespace std;
class Base
{
   int a;
public:
   int b;
protected:
   int c;
public:
Base()
{
a=10;
b=20;
c=30;
}
int returna()
```

www.FirstRanker.com

www.FirstRanker.com

FirstRanker.com
Firstranker's choice
www.FirstRanker.com www.FirstRanker.com


```
{
   return a;
}
};

class child:public Base
{
public:
   void print(Base ob)
   {
   int res=ob.returna();
   cout<<"Accessing data in child class :"<<endl;
   cout<<"Accessing data Private Member by method "<<res<<endl;
   cout<<"Accessing data Public Member of Base class directly "<<b<<endl;
   cout<<"Accessing data Protected Member of Base class directly "<<c<<endl;
   }

};
class child2
{
public:
   void print(Base ob)
   {
   int res=ob.returna();
   cout<<"Accessing data in non-child class :"<<endl;
   cout<<"Accessing data Private Member by method "<<res<<endl;
   cout<<"Accessing data Public Member of Base class by using object "<<ob.b<<endl;
   cout<<"Accessing data Protected Member of Base class not Possible here "<<endl;
   }

};

int main()
{
   Base obj;
   child obj1;
   child2 obj2;
   obj1.print(obj);
   obj2.print(obj);
   return 0;

}
```

www.FirstRanker.com

Output :

```
lab1@lab1:~/Desktop$ g++ Access.cpp
lab1@lab1:~/Desktop$ ./a.out
```


www.FirstRanker.com

FirstRanker.com
Firstranker's choice
www.FirstRanker.com www.FirstRanker.com

Accessing data in child class :
Accessing data Private Member by method 10
Accessing data Public Member of Base class directly 20
Accessing data Protected Member of Base class directly 30
Accessing data in non-child class :
Accessing data Private Member by method 10
Accessing data Public Member of Base class by using object 20
Accessing data Protected Member of Base class not Possible here

Write a program implementing Friend Function

```
#include<iostream>

using namespace std;

class add
{
int a,b;
public:
void setab();
friend void sum(add ab);
};
void add::setab()
{
cout<<"enter a,b"<<endl;
cin>>a;
cin>>b;
}
void sum(add ab)
{
cout<<ab.a+ab.b;
}
int main()
{
add o1;
o1.setab();
sum(o1);
return 0;
}
```

www.FirstRanker.com

Output:

```
lab1@lab1:~/Desktop$ g++ first.cpp

lab1@lab1:~/Desktop$ ./a.out
```


www.FirstRanker.com

FirstRanker.com
Firstranker's choice
www.FirstRanker.com www.FirstRanker.com

enter a,b

10

52

62l

Write a program to illustrate this pointer

```
#include<iostream>

using namespace std;

class jntu
{
int a,b;
public:
jntu(int i,int j)
{
this->a=i;
this->b=j;
}
void display()
{
cout<<this->a<<this->b;
}
};
int main()
{
jntu kkd(3,4);
kkd.display();
return 0;
}
```

www.FirstRanker.com

Output:

lab1@lab1:~/Desktop$ g++ first.cpp

lab1@lab1:~/Desktop$ ./a.out

34

Write a Program to illustrate pointer to a class

```
#include<iostream>
using namespace std;
class Box {
public:
```


www.FirstRanker.com

FirstRanker.com
Firstranker's choice
www.FirstRanker.com www.FirstRanker.com


```
    // Constructor definition
Box(double l = 2.0, double b = 2.0, double h = 2.0) {
cout<<"Constructor called." <<endl;
length = l;
breadth = b;
height = h;
}
double Volume() {
return length * breadth * height;
}
private:
double length;     // Length of a box
double breadth;    // Breadth of a box
double height;     // Height of a box
};
int main(void) {
  Box Box1(3.3, 1.2, 1.5);    // Declare box1
  Box Box2(8.5, 6.0, 2.0);    // Declare box2
  Box *ptrBox;                // Declare pointer to a class.
ptrBox = &Box1;
cout<< "Volume of Box1: " << ptrBox->Volume() << endl;
ptrBox = &Box2;
cout<< "Volume of Box2: " << ptrBox->Volume() << endl;
return 0;
}
```

Output:

```
lab1@lab1:~/Desktop$ g++ first.cpp

lab1@lab1:~/Desktop$ ./a.out

Constructor called.

Constructor called.

Volume of Box1: 5.94

Volume of Box2: 102
```

Exercise -8 (Operator Overloading)

Write a program to Overload Unary, and Binary Operators as Member Function, and Non Member Function.

i.Unary operator as member function

www.FirstRanker.com

www.FirstRanker.com

FirstRanker.com
Firstranker's choice
www.FirstRanker.com www.FirstRanker.com


```
#include<iostream>
using namespace std;
class number
{
float x;
public:
number(float k)
   {x=k;
   }
void operator ++(int)  //postincrement notation
   {
x++;
   }
void operator ++()   //preincrement notation
   {
      ++x;
   }

void show(){cout<<" x="<<x;}
};
int main()
{
```

www.FirstRanker.com

www.FirstRanker.com

FirstRanker.com
Firstranker's choice
www.FirstRanker.com www.FirstRanker.com


```
number N(2.3);
    cout<<" before incrimentation"<<endl;
N.show();
    cout<<" after post incrimentation"<<endl;
    N++;
N.show();
    cout<<" after pre incrementation";
    ++N;
N.show();
return 0;
}
```

Output:

```
lab1@lab1:~/Desktop$ g++ first.cpp
lab1@lab1:~/Desktop$ ./a.out
 before incrimentation
 x=2.3 after post incrimentation
 x=3.3 after pre incrementation x=4.3
```

ii.Binary operator as nonmember function

```
#include<iostream>
using namespace std;
class company
{
int sal;
int nop;
public:
friend void print(company &);
company()
{
sal=0;
```

www.FirstRanker.com

www.FirstRanker.com

FirstRanker.com
Firstranker's choice
www.FirstRanker.com www.FirstRanker.com


```
nop=0;
}
company operator+(company);
company company:: operator-(company c)
{
company temp;
temp.sal=sal-c.sal;
temp.nop=nop-c.nop;
return temp;
}
void set_data(int s, int n)
{
sal=s;
nop=n;
}
};
void print(company &sum)
{
cout<<"Total Salary is :"<<sum.sal<<endl;
cout<<"Total No: of Parts is:"<<sum.nop<<endl;
}
company company:: operator+(company c)
{
company temp;
temp.sal=sal+c.sal;
temp.nop=nop+c.nop;
return temp;
}
void main()
{

int s,n;
company sum,ram,clif,sub;
cout<<"Enter the salary for Ram";
cin>>s;
cout<<"Enter No:Of Parts for Ram:";
cin>>n;
ram.set_data(s,n);
cout<<"Enter the salary for clif";
cin>>s;
cout<<"Enter No:Of Parts for clif:";
cin>>n;
clif.set_data(s,n);
sum=ram+clif;
print(sum);
sub=sum-ram;
```

www.FirstRanker.com

www.FirstRanker.com

FirstRanker.com
Firstranker's choice
www.FirstRanker.com www.FirstRanker.com


```
print(sub);
getch();
}
```

Output:

```
Enter the salary for Ram 5000
Enter No:Of Parts for Ram:2
Enter the salary for clif 3000
Enter No:Of Parts for clif: 4
Total Salary is :8000
Total No: of Parts is:6
Total Salary is :3000
Total No: of Parts is:4
```

Write a c ++ program to implement the overloading assignment = operator

```
#include<iostream>
using namespace std;
#include<iomanip>
class add
{
int a,b;
public:
add() {}
add(int i,int j) {a=i;b=j;}
void display()
{cout<<"a= "<<a<<endl;
cout<<"b= "<<b<<endl;}
add operator=(add a2);
};
add add::operator=( add a2)
{
a=a2.a;
b=a2.b;
return *this;
}
int main()
{
add a1(2,3),a2;
a2=a1;
a2.display();
return 0;
}
```

www.FirstRanker.com

Output:

```
lab1@lab1:~/Desktop$ g++ first.cpp
```


www.FirstRanker.com

FirstRanker.com
Firstranker's choice
www.FirstRanker.com www.FirstRanker.com

lab1@lab1:~/Desktop$ ./a.out

a= 2

b= 3

Write a case study on Overloading Operators and Overloading Functions (150 Words)

C++ allows you to specify more than one definition for a function name or an operator in the same scope, which is called function overloading and operator overloading respectively

When you call an overloaded function or operator, the compiler determines the most appropriate definition to use, by comparing the argument types you have used to call the function or operator with the parameter types specified in the definitions. The process of selecting the most appropriate overloaded function or operator is called overload resolution.

Function Overloading in C++ :

You can have multiple definitions for the same function name in the same scope. The definition of the function must differ from each other by the types and/or the number of arguments in the argument list. You cannot overload function declarations that differ only by return type.

Operators Overloading in C++ :

You can redefine or overload most of the built-in operators available in C++. Thus, a programmer can use operators with user-defined types as well.
Overloaded operators are functions with special names: the keyword "operator" followed by the symbol for the operator being defined. Like any other function, an overloaded operator has a return type and a parameter list.

Ex : Test operator+(const Test&);

www.FirstRanker.com

Exercise -9 (Inheritance)
Write C++ Programs and incorporating various forms of Inheritance
i)Single Inheritance

```
#include<iostream>
using namespace std;
class staff
{
private:
char name[50];
int code;
public:
void getdata();
```


www.FirstRanker.com

FirstRanker.com
Firstranker's choice
www.FirstRanker.com www.FirstRanker.com


```
void display();
};
class typist: public staff
{
private:
int speed;
public:
void getdata();
void display();
};
void staff::getdata()
{
cout<<"Name:";
cin>>name;
cout<<"Code:";
cin>>code;
}
void staff::display()
{
cout<<"Name:"<<name<<endl;
cout<<"Code:"<<code<<endl;
}
void typist::getdata()
{
cout<<"Speed:";
cin>>speed;
}
void typist::display()
{
cout<<"Speed:"<<speed<<endl;
}
int main()
{
typist t;
cout<<"Enter data"<<endl;
t.staff::getdata();
t.getdata();
cout<<endl<<"Display data"<<endl;
t.staff::display();
t.display();
return 0;
}
```

www.FirstRanker.com

Output:

```
lab1@lab1:~/Desktop$ g++ first.cpp
```


www.FirstRanker.com

FirstRanker.com
Firstranker's choice
www.FirstRanker.com www.FirstRanker.com


```
lab1@lab1:~/Desktop$ ./a.out

Enter data

Name:radha

Code:111
Speed:24


Display data

Name:radha

Code:111
Speed:24
```

ii)Hierarchical Inheritance

```
#include<iostream>
using namespace std;
class Number
{
private:
int num;
public:
void getNumber(void)
{
cout<< "Enter an integer number: ";
cin  >> num;
}
//to return num
int returnNumber(void)
{ return num; }
};
//Base Class 1, to calculate square of a number
class Square:public Number
{
public:
int getSquare(void)
{
int num,sqr;
num=returnNumber(); //get number from class Number
sqr=num*num;
return sqr;
```

www.FirstRanker.com

www.FirstRanker.com

FirstRanker.com
Firstranker's choice
www.FirstRanker.com www.FirstRanker.com


```
}
};
//Base Class 2, to calculate cube of a number
class Cube:public Number
{
private:
public:
int getCube(void)
{
int num,cube;
num=returnNumber(); //get number from class Number
cube=num*num*num;
return cube;
}
};
int main()
{
Square objS;
Cube objC;
int sqr,cube;
objS.getNumber();
sqr =objS.getSquare();
cout<< "Square of "<< objS.returnNumber() << " is: " << sqr  << endl;
objC.getNumber();
cube=objC.getCube();
cout<< "Cube   of "<< objS.returnNumber() << " is: " << cube << endl;
return 0;
}
```

Output:

```
lab1@lab1:~/Desktop$ g++ first.cpp

lab1@lab1:~/Desktop$ ./a.out

Enter an integer number: 15

Square of 15 is: 225

Enter an integer number: 22

Cube   of 15 is: 10648
```

iii) Multiple Inheritances

```
#include<iostream>
```

www.FirstRanker.com

www.FirstRanker.com

FirstRanker.com
Firstranker's choice
www.FirstRanker.com www.FirstRanker.com


```
using namespace std;
class base1
{
protected:
int a;
};
class base2
{
protected:
int b;
};
class derived:public base1,public base2
{
int c;
public:
void set(int i,int j)
{a=i;b=j;c=a+b;}
void get()
{
cout<<"a="<<a<<endl<<"b="<<b<<endl<<"c="<<c;
}
};
int main()
{
derived ob1;
ob1.set(10,23);
ob1.get();
return 0;
}
```

Output:

```
lab1@lab1:~/Desktop$ g++ first.cpp

lab1@lab1:~/Desktop$ ./a.out

a=10

b=23

c=33
```

Multi-level inheritance

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

www.FirstRanker.com

www.FirstRanker.com

FirstRanker.com
Firstranker's choice
www.FirstRanker.com www.FirstRanker.com


```
{
protected:
int a,b;
public:
void setbase(int i,int j)
{
a=i;
b=j;
}
void getbase()
{
cout<<"a="<<a<<endl<<"b="<<b<<endl;
}
};
class derived1:public base
{
protected:
int c;
public:
void setd1()
{
c=a+b;
}
void getd1()
{
cout<<"c="<<c<<endl;
}
};
class derived2:public derived1
{
protected:
int d;
public:
void setd2()
{
d=c;
}
void getd2()
{
cout<<"d="<<d<<endl;
}
};
int main()
{
derived2 ob1;
ob1.setbase(10,23);
```

www.FirstRanker.com

www.FirstRanker.com

FirstRanker.com
Firstranker's choice
www.FirstRanker.com www.FirstRanker.com


```
ob1.getbase();
ob1.setd1();
ob1.getd1();
ob1.setd2();
ob1.getd2();
return 0;
}
```

Output:

```
lab1@lab1:~/Desktop$ g++ first.cpp

lab1@lab1:~/Desktop$ ./a.out

a=10

b=23

c=33

d=33
```

Hybrid inheritance

```
#include<iostream>
using namespace std;
class arithmetic
{
protected:
int num1, num2;
public:
void getdata()
{
cout<<"For Addition:";
cout<<"\nEnter the first number: ";
cin>>num1;
cout<<"\nEnter the second number: ";
cin>>num2;
}
};
class plus:public arithmetic
{
protected:
int sum;
public:
void add()
```

www.FirstRanker.com

www.FirstRanker.com

FirstRanker.com
Firstranker's choice
www.FirstRanker.com www.FirstRanker.com


```
{
sum=num1+num2;
}
};
class minus
{
protected:
int n1,n2,diff;
public:
void sub()
{
cout<<"\nFor Subtraction:";
cout<<"\nEnter the first number: ";
cin>>n1;
cout<<"\nEnter the second number: ";
cin>>n2;
diff=n1-n2;
}
};
class result:public plus, public minus
{
public:
void display()
{
cout<<"\nSum of "<<num1<<" and "<<num2<<"= "<<sum;
cout<<"\nDifference of "<<n1<<" and "<<n2<<"= "<<diff;
}
};
void main()
{

result z;
z.getdata();
z.add();
z.sub();
z.display();
getch();
}
```

www.FirstRanker.com

www.FirstRanker.com

FirstRanker.com
Firstranker's choice
www.FirstRanker.com www.FirstRanker.com

Output:

```
For Addition:
Enter the first number: 5

Enter the second number: 6

For Subtraction:
Enter the first number: 3

Enter the second number: 7

Sum of 5 and 6= 11
Difference of 3 and 7= -4_
```

Write a program to show Virtual Base Class

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

class base
//base class derivation
{
  public:
  int i;
};

class derived1 : virtual public base
// derived1 class inherits base class as virtual.
{
  public:
  int j;
};

class derived2 : virtual public base
// derived2 class inherits base class as virtual.
{
  public:
  int k;
};


class derived3 : public derived1, public derived2
/* derived3 class inherits both derived1 and derived2 classes.
This time, there is only one copy of base class in derived3 class,
Because derived2 and derived3 class are inherited virtually.*/
```

www.FirstRanker.com

www.FirstRanker.com

FirstRanker.com
Firstranker's choice
www.FirstRanker.com www.FirstRanker.com


```
{
  public:
  int sum;
};

int main ()
{
        derived3 ob;

  ob.i = 10;
  // accessing base class variable

  ob.j = 20;
  // accessing derived1 class variable

  ob.k = 30;
  // accessing derived2 class variable

  ob.sum = ob.i + ob.j + ob.k;
  // accessing its own original member

  cout << "i = " << ob.i ;
  cout << "\nj = " << ob.j ;
  cout << "\nk = " << ob.k ;
  cout << "\nSum of above variables is : " << ob.sum;

  return 0;
}
```

www.FirstRanker.com

Output:
i = 10
j =20
k = 30
sum of the above variables is 60

Write a case study on using virtual classes (150 Words)
Virtual base classes, used in virtual inheritance, is a way of preventing multiple "instances" of a given class appearing in an inheritance hierarchy when using multiple inheritance

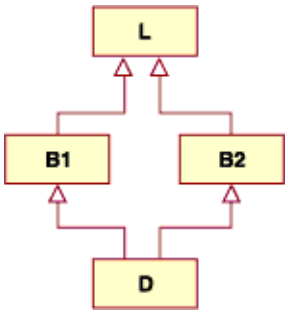
Suppose you have two derived classes B and C that have a common base class A, and you also have another class D that inherits from B and C. You can declare the base class A as *virtual* to ensure that B and C share the same sub object of A.

In the following example, an object of class D has two distinct sub objects of class L, one through class B1 and another through class B2. You can use the keyword virtual in front of the

www.FirstRanker.com

FirstRanker.com
Firstranker's choice
www.FirstRanker.com www.FirstRanker.com

base class specifiers in the *base lists* of classes B1 and B2 to indicate that only one sub object of type L, shared by class B1 and class B2, exists




```
class L { /* ... */ }; // indirect base class
class B1 : virtual public L { /* ... */ };
class B2 : virtual public L { /* ... */ };
class D : public B1, public B2 { /* ... */ }; // valid
```

Exercise-10 (Inheritance –Continued)

Write a Program in C++ to illustrate the order of execution of constructors and destructors in inheritance

```
#include<iostream>

using namespace std;

class Base

{ int x;

public:

Base() { cout << "Base default constructor"; }

};


class Derived : public Base

{ int y;

public:

Derived() { cout << "Derived default constructor"; }

Derived(int i) { cout << "Derived parameterized constructor"; }
```

www.FirstRanker.com

www.FirstRanker.com

FirstRanker.com
Firstranker's choice
www.FirstRanker.com www.FirstRanker.com


```
};


int main()
{
 Base b;
 Derived d1;
 Derived d2(10);
}
```

Output:

```
lab1@lab1:~/Desktop$ g++ first.cpp
lab1@lab1:~/Desktop$ ./a.out

Base default constructor
Base default constructor
Derived default constructor
Base default constructor
Derived parameterized constructor
```

Write a Program to *show* how *constructors* are invoked in *derived class*

```
#include <iostream>
using namespace std;
class A
{
        protected:
                int x;
        public:
                A(int p)
                {
                        x = p;
                }
};
class B : A
{
        private:
                int y;
```

www.FirstRanker.com

www.FirstRanker.com

FirstRanker.com
Firstranker's choice
www.FirstRanker.com www.FirstRanker.com


```
        public:
                B(int p, int q) : A(p)
                {
                        y = q;
                }
                void display()
                {
                        cout<<"x = "<<x<<endl;
                        cout<<"y = "<<y;
                }
};
int main()
{
        B b(10, 20);
        b.display();
        return 0;
}
```

Output

```
x = 10
y = 20
1
2
x = 10
y = 20
```

Exercise -11 (Polymorphism)

Write a program to illustrate runtime polymorphism

```
#include<iostream>

using namespace std;

class Base

{

public:

virtual void show() { cout<<" In Base \n"; }

};



class Derived: public Base
```

www.FirstRanker.com

www.FirstRanker.com

FirstRanker.com
Firstranker's choice
www.FirstRanker.com www.FirstRanker.com


```
{
public:
void show() { cout<<"In Derived \n"; }
};

int main(void)
{
   Base *bp = new Derived;
bp->show();  // RUN-TIME POLYMORPHISM
return 0;
}
```

Output:

```
lab1@lab1:~/Desktop$ g++ first.cpp
lab1@lab1:~/Desktop$ ./a.out
In Derived
```

Write a program to illustrate this pointer

```
#include<iostream>
using namespace std;
class jntu
{
int a,b;
public:
jntu(int i,int j)
{
this->a=i;
this->b=j;
}
void display()
{
cout<<this->a<<this->b;
```

www.FirstRanker.com

www.FirstRanker.com

FirstRanker.com
Firstranker's choice
www.FirstRanker.com www.FirstRanker.com


```
}
};
int main()
{
jntu kkd(3,4);
kkd.display();
return 0;
}
```

Output:

```
lab1@lab1:~/Desktop$ g++ first.cpp

lab1@lab1:~/Desktop$ ./a.out

34
```

Write a program illustrates pure virtual function and calculate the area of different shapes by using abstract class.

```
#include <iostream>
using namespace std;
class Shape
{
        public:
                virtual void area() = 0;
};
class Rectangle : public Shape
{
        private:
                int l;
                int b;
        public:
                Rectangle(int x, int y)
                {
                        l = x;
                        b = y;
                }
                void area()
                {
                        cout<<"Area of rectangle is: "<<(l*b)<<endl;
                }
};
class Circle : public Shape
```

www.FirstRanker.com

www.FirstRanker.com

FirstRanker.com
Firstranker's choice
www.FirstRanker.com www.FirstRanker.com


```
{
        private:
            int r;
        public:
            Circle(int x)
            {
                r = x;
            }
            void area()
            {
                cout<<"Area of circle is: "<<(3.142*r*r)<<endl;
            }
};
int main()
{
        Shape *s;
        s = new Rectangle(10, 20);
        s->area();
        s = new Circle(2);
        s->area();
        return 0;
}
```

Output

Area of rectangle is: 200
Area of circle is: 12.568
Area of rectangle is: 200
Area of circle is: 12.568

www.FirstRanker.com

Write a case study on virtual functions (150 Words)

A virtual function a member function which is declared within base class and is re-defined (Overriden) by derived class.When you refer to a derived class object using a pointer or a reference to the base class, you can call a virtual function for that object and execute the derived class's version of the function.

- Virtual functions ensure that the correct function is called for an object, regardless of the type of reference (or pointer) used for function call.
- They are mainly used to achieve Runtime polymorphism
- Functions are declared with a **virtual** keyword in base class.
- The resolving of function call is done at Run-time.

Rules for Virtual Functions

- They Must be declared in public section of class.
- Virtual functions cannot be static and also cannot be a friend function of another class.

www.FirstRanker.com

FirstRanker.com
Firstranker's choice
www.FirstRanker.com www.FirstRanker.com

- Virtual functions should be accessed using pointer or reference of base class type to achieve run time polymorphism.
- The prototype of virtual functions should be same in base as well as derived class.
- They are always defined in base class and overridden in derived class. It is not mandatory for derived class to override (or re-define the virtual function), in that case base class version of function is used.
- A class may have virtual destructor but it cannot have a virtual constructor.

Exercise -12(Templates)

Write a C++ Program to illustrate template class

```
#include<iostream>
using namespace std;
template<class T>class jntu
{
T a,b;
public:
jntu(T i,T j)
{
a=i;b=j;
}
T add()
{
return a+b;
}
T sub()
{
return a-b;
}
};
int main()
{
jntu<int> hyd(3,4);
cout<<hyd.add()<<endl;
cout<<hyd.sub()<<endl;
jntu<float> hyd2(3.3,4.6);
cout<<hyd2.add()<<endl;
cout<<hyd2.sub()<<endl;
return 0;
}
```

www.FirstRanker.com

Output:

```
lab1@lab1:~/Desktop$ g++ first.cpp

lab1@lab1:~/Desktop$ ./a.out

7
```


www.FirstRanker.com

FirstRanker.com
Firstranker's choice
www.FirstRanker.com www.FirstRanker.com

-1

7.9

-1.3

Write a Program to illustrate class templates with multiple parameters

```
#include<iostream>
using namespace std;
#include<iomanip>
template<class A,class B>class jntu
{
A a;
B b;
        public:
        jntu(A i,B j)
        {
                a=i;b=j;
        }
        void display();
};

template<class A,class B> void jntu<A,B>::display()
{
        cout<<a<<endl<<b<<endl;
}
int main()
{
jntu<int,double> hyd(3,4.5);

hyd.display();

jntu<float,char*> hyd2(3.3,"Jntuh Ds");

hyd2.display();

return 0;
}
```

Output:

```
3
4.5
3.3
Jntuh Ds
```

Write a Program to illustrate member function templates

www.FirstRanker.com

FirstRanker.com
Firstranker's choice
www.FirstRanker.com www.FirstRanker.com


```
#include <iostream>
using namespace std;
  template <typename T>
T myMax(T x, T y)
{
   return (x > y)? x: y;
}

int main()
{
  cout << myMax<int>(3, 7) << endl;  // Call myMax for int
  cout << myMax<double>(3.0, 7.0) << endl; // call myMax for double
  cout << myMax<char>('g', 'e') << endl;   // call myMax for char

  return 0;
}
```

Output:

```
7
7
g
```

www.FirstRanker.com

Exercise -13 (Exception Handling)

Write a Program for Exception Handling Divide by zero

```
#include<iostream>
using namespace std;
void main()
{
   int a,b,c;
   float  d;

   cout<<"Enter the value of a:";
   cin>>a;
   cout<<"Enter the value of b:";
```


www.FirstRanker.com

FirstRanker.com
Firstranker's choice
www.FirstRanker.com www.FirstRanker.com


```
  cin>>b;
  cout<<"Enter the value of c:";
  cin>>c;

  try
  {
        if((a-b)!=0)
        {
          d=c/(a-b);
          cout<<"Result is:"<<d;
        }
        else
        {
          throw(a-b);
        }
  }

  catch(int i)
  {
        cout<<"Answer is infinite because a-b is:"<<i;
  }

  getch();
}
```

Output:

```
        Enter the value for a: 20
        Enter the value for b: 20
        Enter the value for c: 40

        Answer is infinite because a-b is: 0
```

www.FirstRanker.com

Write a Program to rethrow an Exception

```
#include<iostream>
void sub(int j,int k)
{
cout<<"inside function sub()"<<endl;
try
{
if(j==0)
throw j;
else
cout<<"Sunbstraction="<<j-k<<endl;
}
catch(int)
```


www.FirstRanker.com

FirstRanker.com
Firstranker's choice
www.FirstRanker.com www.FirstRanker.com


```
{
cout<<"caught Null Value"<<endl;
throw;
}
cout<<"End of sun()"<<endl;
}
int main()
{
cout<<"indisde function main()"<<endl;
try
{
sub(8,5);
sub(0,8);
}
catch(int)
{
cout<<"Caught null inside main()"<<endl;
}
cout<<"end of function main()"<<endl;
return 0;
}
```

Output:

```
inside function main()
inside function sub()
Substraction=3
End of sub()
Inside function sub()
Caught null value
Caught null inside main()
End of function main()
```

www.FirstRanker.com

Exercise -14 (STL)

Write a Program to implement List and List Operations

```
#include <iostream>
#include <list>
#include <string>
#include <cstdlib>
using namespace std;
int main()
{
   int myints[] = {5, 6, 3, 2, 7};
   list<int> l, l1 (myints, myints + sizeof(myints) / sizeof(int));
   list<int>::iterator it;
   int choice, item;
```


www.FirstRanker.com

FirstRanker.com
Firstranker's choice
www.FirstRanker.com www.FirstRanker.com


```
while (1)
{
   cout<<"\n---------------------"<<endl;
   cout<<"List Implementation in Stl"<<endl;
   cout<<"\n---------------------"<<endl;
   cout<<"1.Insert Element at the Front"<<endl;
   cout<<"2.Insert Element at the End"<<endl;
   cout<<"3.Delete Element at the Front"<<endl;
   cout<<"4.Delete Element at the End"<<endl;
   cout<<"5.Front Element of List"<<endl;
   cout<<"6.Last Element of the List"<<endl;
   cout<<"7.Size of the List"<<endl;
   cout<<"8.Resize List"<<endl;
   cout<<"9.Remove Elements with Specific Values"<<endl;
   cout<<"10.Remove Duplicate Values"<<endl;
   cout<<"11.Reverse the order of elements"<<endl;
   cout<<"12.Sort Forward List"<<endl;
   cout<<"13.Merge Sorted Lists"<<endl;
   cout<<"14.Display Forward List"<<endl;
   cout<<"15.Exit"<<endl;
   cout<<"Enter your Choice: ";
   cin>>choice;
   switch(choice)
   {
   case 1:
      cout<<"Enter value to be inserted at the front: ";
      cin>>item;
      l.push_front(item);
      break;
   case 2:
      cout<<"Enter value to be inserted at the end: ";
      cin>>item;
      l.push_back(item);
      break;
   case 3:
      item = l.front();
      l.pop_front();
            cout<<"Element "<<item<<" deleted"<<endl;
      break;
   case 4:
      item = l.back();
      l.pop_back();
            cout<<"Element "<<item<<" deleted"<<endl;
      break;
   case 5:
            cout<<"Front Element of the List: ";
```

www.FirstRanker.com

www.FirstRanker.com

FirstRanker.com
Firstranker's choice
www.FirstRanker.com www.FirstRanker.com


```
            cout<<l.front()<<endl;
        break;
    case 6:
        cout<<"Last Element of the List: ";
        cout<<l.back()<<endl;
        break;
    case 7:
        cout<<"Size of the List: "<<l.size()<<endl;
        break;
    case 8:
        cout<<"Enter new size of the List: ";
        cin>>item;
        if (item <= l.size())
            l.resize(item);
        else
            l.resize(item, 0);
        break;
    case 9:
        cout<<"Enter element to be deleted: ";
        cin>>item;
        l.remove(item);
        break;
    case 10:
        l.unique();
        cout<<"Duplicate Items Deleted"<<endl;
        break;
    case 11:
        l.reverse();
        cout<<"List reversed"<<endl;
        break;
    case 12:
        l.sort();
        cout<<"List Sorted"<<endl;
        break;
    case 13:
        l1.sort();
        l.sort();
        l.merge(l1);
        cout<<"Lists Merged after sorting"<<endl;
        break;
    case 14:
        cout<<"Elements of the List:  ";
        for (it = l.begin(); it != l.end(); it++)
            cout<<*it<<"  ";
        cout<<endl;
        break;
```

www.FirstRanker.com

www.FirstRanker.com

FirstRanker.com
Firstranker's choice
www.FirstRanker.com www.FirstRanker.com


```
                case 15:
          exit(1);
                    break;
       default:
          cout<<"Wrong Choice"<<endl;
       }
   }
   return 0;
}
```

Output:

```
$ g++ list.cpp
$ a.out
---------------------
List Implementation in Stl

---------------------
1.Insert Element at the Front
2.Insert Element at the End
3.Delete Element at the Front
4.Delete Element at the End
5.Front Element of List
6.Last Element of the List
7.Size of the List
8.Resize List
9.Remove Elements with Specific Values
10.Remove Duplicate Values
11.Reverse the order of elements
12.Sort Forward List
13.Merge Sorted Lists
14.Display Forward List
15.Exit
Enter your Choice: 1
Enter value to be inserted at the front: 5

---------------------
List Implementation in Stl

---------------------
1.Insert Element at the Front
2.Insert Element at the End
3.Delete Element at the Front
4.Delete Element at the End
5.Front Element of List
6.Last Element of the List
7.Size of the List
```

www.FirstRanker.com

www.FirstRanker.com

FirstRanker.com
Firstranker's choice
www.FirstRanker.com www.FirstRanker.com

8.Resize List
9.Remove Elements with Specific Values
10.Remove Duplicate Values
11.Reverse the order of elements
12.Sort Forward List
13.Merge Sorted Lists
14.Display Forward List
15.Exit
Enter your Choice: 1
Enter value to be inserted at the front: 6

List Implementation in Stl

1.Insert Element at the Front
2.Insert Element at the End
3.Delete Element at the Front
4.Delete Element at the End
5.Front Element of List
6.Last Element of the List
7.Size of the List
8.Resize List
9.Remove Elements with Specific Values
10.Remove Duplicate Values
11.Reverse the order of elements
12.Sort Forward List
13.Merge Sorted Lists
14.Display Forward List
15.Exit
Enter your Choice: 1
Enter value to be inserted at the front: 7

www.FirstRanker.com

List Implementation in Stl

1.Insert Element at the Front
2.Insert Element at the End
3.Delete Element at the Front
4.Delete Element at the End
5.Front Element of List
6.Last Element of the List
7.Size of the List
8.Resize List
9.Remove Elements with Specific Values

www.FirstRanker.com

FirstRanker.com
Firstranker's choice
www.FirstRanker.com www.FirstRanker.com

10.Remove Duplicate Values
11.Reverse the order of elements
12.Sort Forward List
13.Merge Sorted Lists
14.Display Forward List
15.Exit
Enter your Choice: 1
Enter value to be inserted at the front: 8

List Implementation in Stl

1.Insert Element at the Front
2.Insert Element at the End
3.Delete Element at the Front
4.Delete Element at the End
5.Front Element of List
6.Last Element of the List
7.Size of the List
8.Resize List
9.Remove Elements with Specific Values
10.Remove Duplicate Values
11.Reverse the order of elements
12.Sort Forward List
13.Merge Sorted Lists
14.Display Forward List
15.Exit
Enter your Choice: 2
Enter value to be inserted at the end: 4

www.FirstRanker.com

List Implementation in Stl

1.Insert Element at the Front
2.Insert Element at the End
3.Delete Element at the Front
4.Delete Element at the End
5.Front Element of List
6.Last Element of the List
7.Size of the List
8.Resize List
9.Remove Elements with Specific Values
10.Remove Duplicate Values
11.Reverse the order of elements

www.FirstRanker.com

FirstRanker.com
Firstranker's choice
www.FirstRanker.com www.FirstRanker.com

12.Sort Forward List
13.Merge Sorted Lists
14.Display Forward List
15.Exit
Enter your Choice: 2
Enter value to be inserted at the end: 3

List Implementation in Stl

1.Insert Element at the Front
2.Insert Element at the End
3.Delete Element at the Front
4.Delete Element at the End
5.Front Element of List
6.Last Element of the List
7.Size of the List
8.Resize List
9.Remove Elements with Specific Values
10.Remove Duplicate Values
11.Reverse the order of elements
12.Sort Forward List
13.Merge Sorted Lists
14.Display Forward List
15.Exit
Enter your Choice: 2
Enter value to be inserted at the end: 2

www.FirstRanker.com

List Implementation in Stl

1.Insert Element at the Front
2.Insert Element at the End
3.Delete Element at the Front
4.Delete Element at the End
5.Front Element of List
6.Last Element of the List
7.Size of the List
8.Resize List
9.Remove Elements with Specific Values
10.Remove Duplicate Values
11.Reverse the order of elements
12.Sort Forward List
13.Merge Sorted Lists

www.FirstRanker.com

FirstRanker.com
Firstranker's choice
www.FirstRanker.com www.FirstRanker.com

14.Display Forward List
15.Exit
Enter your Choice: 7
Size of the List: 7

List Implementation in Stl

1.Insert Element at the Front
2.Insert Element at the End
3.Delete Element at the Front
4.Delete Element at the End
5.Front Element of List
6.Last Element of the List
7.Size of the List
8.Resize List
9.Remove Elements with Specific Values
10.Remove Duplicate Values
11.Reverse the order of elements
12.Sort Forward List
13.Merge Sorted Lists
14.Display Forward List
15.Exit
Enter your Choice: 14
Elements of the List: 8 7 6 5 4 3 2

List Implementation in Stl

1.Insert Element at the Front
2.Insert Element at the End
3.Delete Element at the Front
4.Delete Element at the End
5.Front Element of List
6.Last Element of the List
7.Size of the List
8.Resize List
9.Remove Elements with Specific Values
10.Remove Duplicate Values
11.Reverse the order of elements
12.Sort Forward List
13.Merge Sorted Lists
14.Display Forward List
15.Exit

www.FirstRanker.com

www.FirstRanker.com

FirstRanker.com
Firstranker's choice
www.FirstRanker.com www.FirstRanker.com

Enter your Choice: 5
Front Element of the List: 8

List Implementation in Stl

1.Insert Element at the Front
2.Insert Element at the End
3.Delete Element at the Front
4.Delete Element at the End
5.Front Element of List
6.Last Element of the List
7.Size of the List
8.Resize List
9.Remove Elements with Specific Values
10.Remove Duplicate Values
11.Reverse the order of elements
12.Sort Forward List
13.Merge Sorted Lists
14.Display Forward List
15.Exit
Enter your Choice: 6
Last Element of the List: 2

List Implementation in Stl

1.Insert Element at the Front
2.Insert Element at the End
3.Delete Element at the Front
4.Delete Element at the End
5.Front Element of List
6.Last Element of the List
7.Size of the List
8.Resize List
9.Remove Elements with Specific Values
10.Remove Duplicate Values
11.Reverse the order of elements
12.Sort Forward List
13.Merge Sorted Lists
14.Display Forward List
15.Exit
Enter your Choice: 9
Enter element to be deleted: 7

www.FirstRanker.com

www.FirstRanker.com

FirstRanker.com
Firstranker's choice
www.FirstRanker.com www.FirstRanker.com

List Implementation in Stl

1.Insert Element at the Front
2.Insert Element at the End
3.Delete Element at the Front
4.Delete Element at the End
5.Front Element of List
6.Last Element of the List
7.Size of the List
8.Resize List
9.Remove Elements with Specific Values
10.Remove Duplicate Values
11.Reverse the order of elements
12.Sort Forward List
13.Merge Sorted Lists
14.Display Forward List
15.Exit
Enter your Choice: 14
Elements of the List: 8 6 5 4 3 2

List Implementation in Stl

1.Insert Element at the Front
2.Insert Element at the End
3.Delete Element at the Front
4.Delete Element at the End
5.Front Element of List
6.Last Element of the List
7.Size of the List
8.Resize List
9.Remove Elements with Specific Values
10.Remove Duplicate Values
11.Reverse the order of elements
12.Sort Forward List
13.Merge Sorted Lists
14.Display Forward List
15.Exit
Enter your Choice: 1
Enter value to be inserted at the front: 8

---

www.FirstRanker.com

www.FirstRanker.com

FirstRanker.com
Firstranker's choice
www.FirstRanker.com www.FirstRanker.com

List Implementation in Stl

1.Insert Element at the Front
2.Insert Element at the End
3.Delete Element at the Front
4.Delete Element at the End
5.Front Element of List
6.Last Element of the List
7.Size of the List
8.Resize List
9.Remove Elements with Specific Values
10.Remove Duplicate Values
11.Reverse the order of elements
12.Sort Forward List
13.Merge Sorted Lists
14.Display Forward List
15.Exit
Enter your Choice: 2
Enter value to be inserted at the end: 2

List Implementation in Stl

1.Insert Element at the Front
2.Insert Element at the End
3.Delete Element at the Front
4.Delete Element at the End
5.Front Element of List
6.Last Element of the List
7.Size of the List
8.Resize List
9.Remove Elements with Specific Values
10.Remove Duplicate Values
11.Reverse the order of elements
12.Sort Forward List
13.Merge Sorted Lists
14.Display Forward List
15.Exit
Enter your Choice: 14
Elements of the List: 8 8 6 5 4 3 2 2

List Implementation in Stl

www.FirstRanker.com

www.FirstRanker.com

FirstRanker.com
Firstranker's choice
www.FirstRanker.com www.FirstRanker.com

1.Insert Element at the Front
2.Insert Element at the End
3.Delete Element at the Front
4.Delete Element at the End
5.Front Element of List
6.Last Element of the List
7.Size of the List
8.Resize List
9.Remove Elements with Specific Values
10.Remove Duplicate Values
11.Reverse the order of elements
12.Sort Forward List
13.Merge Sorted Lists
14.Display Forward List
15.Exit
Enter your Choice: 10
Duplicate Items Deleted

List Implementation in Stl

1.Insert Element at the Front
2.Insert Element at the End
3.Delete Element at the Front
4.Delete Element at the End
5.Front Element of List
6.Last Element of the List
7.Size of the List
8.Resize List
9.Remove Elements with Specific Values
10.Remove Duplicate Values
11.Reverse the order of elements
12.Sort Forward List
13.Merge Sorted Lists
14.Display Forward List
15.Exit
Enter your Choice: 14
Elements of the List: 8 6 5 4 3 2

List Implementation in Stl

1.Insert Element at the Front

www.FirstRanker.com

www.FirstRanker.com

FirstRanker.com
Firstranker's choice
www.FirstRanker.com www.FirstRanker.com

2.Insert Element at the End
3.Delete Element at the Front
4.Delete Element at the End
5.Front Element of List
6.Last Element of the List
7.Size of the List
8.Resize List
9.Remove Elements with Specific Values
10.Remove Duplicate Values
11.Reverse the order of elements
12.Sort Forward List
13.Merge Sorted Lists
14.Display Forward List
15.Exit
Enter your Choice: 12
List Sorted

List Implementation in Stl

1.Insert Element at the Front
2.Insert Element at the End
3.Delete Element at the Front
4.Delete Element at the End
5.Front Element of List
6.Last Element of the List
7.Size of the List
8.Resize List
9.Remove Elements with Specific Values
10.Remove Duplicate Values
11.Reverse the order of elements
12.Sort Forward List
13.Merge Sorted Lists
14.Display Forward List
15.Exit
Enter your Choice: 14
Elements of the List: 2 3 4 5 6 8

List Implementation in Stl

1.Insert Element at the Front
2.Insert Element at the End
3.Delete Element at the Front

www.FirstRanker.com

www.FirstRanker.com

FirstRanker.com
Firstranker's choice
www.FirstRanker.com www.FirstRanker.com

4.Delete Element at the End
5.Front Element of List
6.Last Element of the List
7.Size of the List
8.Resize List
9.Remove Elements with Specific Values
10.Remove Duplicate Values
11.Reverse the order of elements
12.Sort Forward List
13.Merge Sorted Lists
14.Display Forward List
15.Exit
Enter your Choice: 13
Lists Merged after sorting

List Implementation in Stl

1.Insert Element at the Front
2.Insert Element at the End
3.Delete Element at the Front
4.Delete Element at the End
5.Front Element of List
6.Last Element of the List
7.Size of the List
8.Resize List
9.Remove Elements with Specific Values
10.Remove Duplicate Values
11.Reverse the order of elements
12.Sort Forward List
13.Merge Sorted Lists
14.Display Forward List
15.Exit
Enter your Choice: 14
Elements of the List: 2 2 3 3 4 5 5 6 6 7 8

List Implementation in Stl

1.Insert Element at the Front
2.Insert Element at the End
3.Delete Element at the Front
4.Delete Element at the End
5.Front Element of List

www.FirstRanker.com

www.FirstRanker.com

FirstRanker.com
Firstranker's choice
www.FirstRanker.com www.FirstRanker.com

6.Last Element of the List
7.Size of the List
8.Resize List
9.Remove Elements with Specific Values
10.Remove Duplicate Values
11.Reverse the order of elements
12.Sort Forward List
13.Merge Sorted Lists
14.Display Forward List
15.Exit
Enter your Choice: 9
Enter element to be deleted: 7

List Implementation in Stl

1.Insert Element at the Front
2.Insert Element at the End
3.Delete Element at the Front
4.Delete Element at the End
5.Front Element of List
6.Last Element of the List
7.Size of the List
8.Resize List
9.Remove Elements with Specific Values
10.Remove Duplicate Values
11.Reverse the order of elements
12.Sort Forward List
13.Merge Sorted Lists
14.Display Forward List
15.Exit
Enter your Choice: 14
Elements of the List: 2 2 3 3 4 5 5 6 6 8

List Implementation in Stl

1.Insert Element at the Front
2.Insert Element at the End
3.Delete Element at the Front
4.Delete Element at the End
5.Front Element of List
6.Last Element of the List
7.Size of the List

www.FirstRanker.com

www.FirstRanker.com

FirstRanker.com
Firstranker's choice
www.FirstRanker.com www.FirstRanker.com

8.Resize List
9.Remove Elements with Specific Values
10.Remove Duplicate Values
11.Reverse the order of elements
12.Sort Forward List
13.Merge Sorted Lists
14.Display Forward List
15.Exit
Enter your Choice: 11
List reversed

List Implementation in Stl

1.Insert Element at the Front
2.Insert Element at the End
3.Delete Element at the Front
4.Delete Element at the End
5.Front Element of List
6.Last Element of the List
7.Size of the List
8.Resize List
9.Remove Elements with Specific Values
10.Remove Duplicate Values
11.Reverse the order of elements
12.Sort Forward List
13.Merge Sorted Lists
14.Display Forward List
15.Exit
Enter your Choice: 14
Elements of the List: 8 6 6 5 5 4 3 3 2 2

www.FirstRanker.com

List Implementation in Stl

1.Insert Element at the Front
2.Insert Element at the End
3.Delete Element at the Front
4.Delete Element at the End
5.Front Element of List
6.Last Element of the List
7.Size of the List
8.Resize List
9.Remove Elements with Specific Values

www.FirstRanker.com

FirstRanker.com
Firstranker's choice
www.FirstRanker.com www.FirstRanker.com

10.Remove Duplicate Values
11.Reverse the order of elements
12.Sort Forward List
13.Merge Sorted Lists
14.Display Forward List
15.Exit
Enter your Choice: 10
Duplicate Items Deleted

List Implementation in Stl

1.Insert Element at the Front
2.Insert Element at the End
3.Delete Element at the Front
4.Delete Element at the End
5.Front Element of List
6.Last Element of the List
7.Size of the List
8.Resize List
9.Remove Elements with Specific Values
10.Remove Duplicate Values
11.Reverse the order of elements
12.Sort Forward List
13.Merge Sorted Lists
14.Display Forward List
15.Exit
Enter your Choice: 14
Elements of the List: 8 6 5 4 3 2

List Implementation in Stl

1.Insert Element at the Front
2.Insert Element at the End
3.Delete Element at the Front
4.Delete Element at the End
5.Front Element of List
6.Last Element of the List
7.Size of the List
8.Resize List
9.Remove Elements with Specific Values
10.Remove Duplicate Values
11.Reverse the order of elements

www.FirstRanker.com

www.FirstRanker.com

FirstRanker.com
Firstranker's choice
www.FirstRanker.com www.FirstRanker.com

12.Sort Forward List
13.Merge Sorted Lists
14.Display Forward List
15.Exit
Enter your Choice: 15

Write a Program to implementVector andVector Operations

```
#include <iostream>
#include <vector>
#include <string>
#include <cstdlib>
using namespace std;
int main()
{
   vector<int> ss;
   vector<int>::iterator it;
   int choice, item;
   while (1)
   {
      cout<<"\n----------------------"<<endl;
      cout<<"Vector Implementation in Stl"<<endl;
      cout<<"\n----------------------"<<endl;
      cout<<"1.Insert Element into the Vector"<<endl;
      cout<<"2.Delete Last Element of the Vector"<<endl;
      cout<<"3.Size of the Vector"<<endl;
      cout<<"4.Display by Index"<<endl;
      cout<<"5.Dislplay by Iterator"<<endl;
      cout<<"6.Clear the Vector"<<endl;
      cout<<"7.Exit"<<endl;
      cout<<"Enter your Choice: ";
      cin>>choice;
      switch(choice)
      {
      case 1:
         cout<<"Enter value to be inserted: ";
         cin>>item;
         ss.push_back(item);
         break;
      case 2:
         cout<<"Delete Last Element Inserted:"<<endl;
         ss.pop_back();
         break;
      case 3:
         cout<<"Size of Vector: ";
         cout<<ss.size()<<endl;
```

www.FirstRanker.com

www.FirstRanker.com

FirstRanker.com
Firstranker's choice
www.FirstRanker.com www.FirstRanker.com


```
            break;
        case 4:
            cout<<"Displaying Vector by Index: ";
            for (int i = 0; i < ss.size(); i++)
            {
                cout<<ss[i]<<" ";
            }
            cout<<endl;
            break;
        case 5:
            cout<<"Displaying Vector by Iterator: ";
            for (it = ss.begin(); it != ss.end(); it++)
            {
                cout<<*it<<" ";
            }
            cout<<endl;
            break;
        case 6:
            ss.clear();
            cout<<"Vector Cleared"<<endl;
            break;
        case 7:
            exit(1);
            break;
        default:
            cout<<"Wrong Choice"<<endl;
        }
    }
    return 0;
}
```

Output:

```
$ g++ vector.cpp
$ a.out
---------------------
Vector Implementation in Stl

---------------------
1.Insert Element into the Vector
2.Delete Last Element of the Vector
3.Size of the Vector
4.Display by Index
5.Dislplay by Iterator
6.Clear the Vector
7.Exit
```

www.FirstRanker.com

www.FirstRanker.com

FirstRanker.com
Firstranker's choice
www.FirstRanker.com www.FirstRanker.com


```
Enter your Choice: 1
Enter value to be inserted: 4

---------------------
Vector Implementation in Stl

---------------------
1.Insert Element into the Vector
2.Delete Last Element of the Vector
3.Size of the Vector
4.Display by Index
5.Dislplay by Iterator
6.Clear the Vector
7.Exit
Enter your Choice: 1
Enter value to be inserted: 6

---------------------
Vector Implementation in Stl

---------------------
1.Insert Element into the Vector
2.Delete Last Element of the Vector
3.Size of the Vector
4.Display by Index
5.Dislplay by Iterator
6.Clear the Vector
7.Exit
Enter your Choice: 1
Enter value to be inserted: 3

---------------------
Vector Implementation in Stl

---------------------
1.Insert Element into the Vector
2.Delete Last Element of the Vector
3.Size of the Vector
4.Display by Index
5.Dislplay by Iterator
6.Clear the Vector
7.Exit
Enter your Choice: 1
Enter value to be inserted: 8

---------------------
```

www.FirstRanker.com

www.FirstRanker.com

FirstRanker.com
Firstranker's choice
www.FirstRanker.com www.FirstRanker.com

Vector Implementation in Stl

1.Insert Element into the Vector
2.Delete Last Element of the Vector
3.Size of the Vector
4.Display by Index
5.Dislplay by Iterator
6.Clear the Vector
7.Exit
Enter your Choice: 1
Enter value to be inserted: 9

Vector Implementation in Stl

1.Insert Element into the Vector
2.Delete Last Element of the Vector
3.Size of the Vector
4.Display by Index
5.Dislplay by Iterator
6.Clear the Vector
7.Exit
Enter your Choice: 1
Enter value to be inserted: 2

Vector Implementation in Stl

1.Insert Element into the Vector
2.Delete Last Element of the Vector
3.Size of the Vector
4.Display by Index
5.Dislplay by Iterator
6.Clear the Vector
7.Exit
Enter your Choice: 3
Size of Vector: 6

Vector Implementation in Stl

1.Insert Element into the Vector

www.FirstRanker.com

www.FirstRanker.com

FirstRanker.com
Firstranker's choice
www.FirstRanker.com www.FirstRanker.com

2.Delete Last Element of the Vector
3.Size of the Vector
4.Display by Index
5.Dislplay by Iterator
6.Clear the Vector
7.Exit
Enter your Choice: 4
Displaying Vector by Index: 4 6 3 8 9 2

Vector Implementation in Stl

1.Insert Element into the Vector
2.Delete Last Element of the Vector
3.Size of the Vector
4.Display by Index
5.Dislplay by Iterator
6.Clear the Vector
7.Exit
Enter your Choice: 2
Delete Last Element Inserted:

Vector Implementation in Stl

1.Insert Element into the Vector
2.Delete Last Element of the Vector
3.Size of the Vector
4.Display by Index
5.Dislplay by Iterator
6.Clear the Vector
7.Exit
Enter your Choice: 3
Size of Vector: 5

Vector Implementation in Stl

1.Insert Element into the Vector
2.Delete Last Element of the Vector
3.Size of the Vector
4.Display by Index
5.Dislplay by Iterator

www.FirstRanker.com

www.FirstRanker.com

FirstRanker.com
Firstranker's choice
www.FirstRanker.com www.FirstRanker.com

6.Clear the Vector
7.Exit
Enter your Choice: 5
Displaying Vector by Iterator: 4 6 3 8 9

Vector Implementation in Stl

1.Insert Element into the Vector
2.Delete Last Element of the Vector
3.Size of the Vector
4.Display by Index
5.Dislplay by Iterator
6.Clear the Vector
7.Exit
Enter your Choice: 4
Displaying Vector by Index: 4 6 3 8 9

Vector Implementation in Stl

1.Insert Element into the Vector
2.Delete Last Element of the Vector
3.Size of the Vector
4.Display by Index
5.Dislplay by Iterator
6.Clear the Vector
7.Exit
Enter your Choice: 6
Vector Cleared

www.FirstRanker.com

Vector Implementation in Stl

1.Insert Element into the Vector
2.Delete Last Element of the Vector
3.Size of the Vector
4.Display by Index
5.Dislplay by Iterator
6.Clear the Vector
7.Exit
Enter your Choice: 3
Size of Vector: 0

www.FirstRanker.com

FirstRanker.com
Firstranker's choice
www.FirstRanker.com www.FirstRanker.com

Vector Implementation in Stl

1.Insert Element into the Vector
2.Delete Last Element of the Vector
3.Size of the Vector
4.Display by Index
5.Dislplay by Iterator
6.Clear the Vector
7.Exit
Enter your Choice: 7

Exercise -15 (STLContinued)
Write a Program to implement Deque and Deque Operations

```
#include <iostream>
#include <cstdlib>
using namespace std;
/*
 * Node Declaration
 */
struct node
{
   int info;
   node *next;
   node *prev;

}*head, *tail;

/*
 * Class Declaration
 */
class dqueue
{
   public:
      int top1, top2;
      void insert();
      void del();
      void display();
      dqueue()
      {
         top1 = 0;
         top2 = 0;
         head = NULL;
```

www.FirstRanker.com

www.FirstRanker.com

FirstRanker.com
Firstranker's choice
www.FirstRanker.com www.FirstRanker.com


```
            tail = NULL;
        }
};

/*
 * Main: Contains Menu
 */
int main()
{
    int choice;
    dqueue dl;
    while (1)
    {
        cout<<"\n-------------"<<endl;
        cout<<"Operations on Deque"<<endl;
        cout<<"\n-------------"<<endl;
        cout<<"1.Insert Element into the Deque"<<endl;
        cout<<"2.Delete Element from the Deque"<<endl;
        cout<<"3.Traverse the Deque"<<endl;
        cout<<"4.Quit"<<endl;
        cout<<"Enter your Choice: ";
        cin>>choice;
        cout<<endl;
        switch(choice)
        {
        case 1:
            dl.insert();
            break;
        case 2:
            dl.del();
            break;
        case 3:
            dl.display();
            break;
        case 4:
            exit(1);
            break;
        default:
            cout<<"Wrong Choice"<<endl;
        }
    }
    return 0;
}

/*
 * Insert Element in Doubly Ended Queue
```

www.FirstRanker.com

www.FirstRanker.com

FirstRanker.com
Firstranker's choice
www.FirstRanker.com www.FirstRanker.com


```
 */
void dqueue::insert()
{
   struct node *temp;
   int ch, value;
   if (top1 + top2 >= 50)
   {
      cout<<"Dequeue Overflow"<<endl;
      return;
   }
   if (top1 + top2 == 0)
   {
      cout<<"Enter the value to be inserted: ";
      cin>>value;
      head = new (struct node);
      head->info = value;
      head->next = NULL;
      head->prev = NULL;
      tail = head;
      top1++;
      cout<<"Element Inserted into empty deque"<<endl;
   }
   else
   {
      while (1)
      {
         cout<<endl;
         cout<<"1.Insert Element at first"<<endl;
         cout<<"2.Insert Element at last"<<endl;
         cout<<"3.Exit"<<endl;
         cout<<endl;
         cout<<"Enter Your Choice: ";
         cin>>ch;
         cout<<endl;
         switch(ch)
         {
         case 1:
            cout<<"Enter the value to be inserted: ";
            cin>>value;
            temp = new (struct node);
            temp->info = value;
            temp->next = head;
            temp->prev = NULL;
            head->prev = temp;
            head = temp;
            top1++;
```

www.FirstRanker.com

www.FirstRanker.com

FirstRanker.com
Firstranker's choice
www.FirstRanker.com www.FirstRanker.com


```
            break;
        case 2:
            cout<<"Enter the value to be inserted: ";
            cin>>value;
            temp = new (struct node);
            temp->info = value;
            temp->next = NULL;
            temp->prev = tail;
            tail->next = temp;
            tail = temp;
            top2++;
            break;
        case 3:
            return;
            break;
        default:
            cout<<"Wrong Choice"<<endl;
        }
    }
  }
}

/*
 * Delete Element in Doubly Ended Queue
 */
void dqueue::del()
{
   if (top1 + top2 <= 0)
   {
      cout<<"Deque Underflow"<<endl;
      return;
   }
   int ch;
   while (1)
   {
      cout<<endl;
      cout<<"1.Delete Element at first"<<endl;
      cout<<"2.Delete Element at last"<<endl;
      cout<<"3.Exit"<<endl;
      cout<<endl;
      cout<<"Enter Your Choice: ";
      cin>>ch;
      cout<<endl;
      switch(ch)
      {
      case 1:
```

www.FirstRanker.com

www.FirstRanker.com

FirstRanker.com
Firstranker's choice
www.FirstRanker.com www.FirstRanker.com


```
            head = head->next;
            head->prev = NULL;
            top1--;
            break;
        case 2:
            tail = tail->prev;
            tail->next = NULL;
            top2--;
            break;
        case 3:
            return;
            break;
        default:
            cout<<"Wrong Choice"<<endl;
        }
    }
}

/*
 * Display Doubly Ended Queue
 */
void dqueue::display()
{
    struct node *temp;
    int ch;
    if (top1 + top2 <= 0)
    {
        cout<<"Deque Underflow"<<endl;
        return;
    }
    while (1)
    {
        cout<<endl;
        cout<<"1.Display Deque from Beginning"<<endl;
        cout<<"2.Display Deque from End"<<endl;
        cout<<"3.Exit"<<endl;
        cout<<endl;
        cout<<"Enter Your Choice: ";
        cin>>ch;
        cout<<endl;
        switch (ch)
        {
        case 1:
            temp = head;
            cout<<"Deque from Beginning:"<<endl;
            while (temp != NULL)
```

www.FirstRanker.com

www.FirstRanker.com

FirstRanker.com
Firstranker's choice
www.FirstRanker.com www.FirstRanker.com


```
        {
           cout<<temp->info<<" ";
           temp = temp->next;
        }
        cout<<endl;
        break;
     case 2:
        cout<<"Deque from End:"<<endl;
        temp = tail;
        while (temp != NULL)
        {
           cout<<temp->info<<" ";
           temp = temp->prev;
        }
        temp = tail;
        cout<<endl;
        break;
     case 3:
        return;
        break;
     default:
        cout<<"Wrong Choice"<<endl;
     }
  }
}
```

Output:

```
Enter data Of New Node (Enter 0 To Have other options):1

Enter data Of New Node (Enter 0 To Have other options):2

Enter data Of New Node (Enter 0 To Have other options):3

Enter data Of New Node (Enter 0 To Have other options):4

Enter data Of New Node (Enter 0 To Have other options):0

 1.Insertion
2.Deletion
3.Print List
4.Exit

 Enter your option : 3
1   2   3   4
 1.Insertion
2.Deletion
3.Print List
4.Exit

 Enter your option : 4
```


www.FirstRanker.com

FirstRanker.com
Firstranker's choice
www.FirstRanker.com www.FirstRanker.com

Write a Program to implement Map and Map Operations

```
#include <iostream>
#include <map>
#include <string>
#include <cstdlib>
using namespace std;
int main()
{
   map<char,int> mp;
   map<char, int>::iterator it;
   int choice, item;
   char s;
   while (1)
   {
      cout<<"\n----------------------"<<endl;
      cout<<"Map Implementation in Stl"<<endl;
      cout<<"\n----------------------"<<endl;
      cout<<"1.Insert Element into the Map"<<endl;
      cout<<"2.Delete Element of the Map"<<endl;
               cout<<"3.Size of the Map"<<endl;
      cout<<"4.Find Element at a key in Map"<<endl;
      cout<<"5.Dislplay by Iterator"<<endl;
      cout<<"6.Count Elements at a specific key"<<endl;
      cout<<"7.Exit"<<endl;
      cout<<"Enter your Choice: ";
      cin>>choice;
      switch(choice)
      {
      case 1:
         cout<<"Enter value to be inserted: ";
         cin>>item;
         cout<<"Enter the key: ";
         cin>>s;
         mp.insert(pair<char,int>(s  ,item));
         break;
      case 2:
         cout<<"Enter the mapped string to be deleted: ";
         cin>>s;
         mp.erase(s);
         break;
      case 3:
                  cout<<"Size of Map: ";
                  cout<<mp.size()<<endl;
```

www.FirstRanker.com

www.FirstRanker.com

FirstRanker.com
Firstranker's choice
www.FirstRanker.com www.FirstRanker.com


```
        break;
    case 4:
                cout<<"Enter the key at which value to be found: ";
        cin>>s;
        if (mp.count(s) != 0)
            cout<<mp.find(s)->second<<endl;
        else
            cout<<"No Element Found"<<endl;
        break;
    case 5:
                cout<<"Displaying Map by Iterator: ";
        for (it = mp.begin(); it != mp.end(); it++)
        {
            cout << (*it).first << ": " << (*it).second << endl;
        }
        break;
    case 6:
        cout<<"Enter the key at which number of values to be counted: ";
        cin>>s;
        cout<<mp.count(s)<<endl;
        break;
              case 7:
        exit(1);
                break;
    default:
        cout<<"Wrong Choice"<<endl;
    }
  }
  return 0;
}
```

Output:

```
$ g++ map.cpp
$ a.out
---------------------
Map Implementation in Stl

---------------------
1.Insert Element into the Map
2.Delete Element of the Map
3.Size of the Map
4.Find Element at a key in Map
5.Dislplay by Iterator
6.Count Elements at a specific key
7.Exit
```

www.FirstRanker.com

www.FirstRanker.com

FirstRanker.com
Firstranker's choice
www.FirstRanker.com www.FirstRanker.com


```
Enter your Choice: 1
Enter value to be inserted: 1
Enter the key: a

---------------------
Map Implementation in Stl

---------------------
1.Insert Element into the Map
2.Delete Element of the Map
3.Size of the Map
4.Find Element at a key in Map
5.Dislplay by Iterator
6.Count Elements at a specific key
7.Exit
Enter your Choice: 1
Enter value to be inserted: 2
Enter the key: b

---------------------
Map Implementation in Stl

---------------------
1.Insert Element into the Map
2.Delete Element of the Map
3.Size of the Map
4.Find Element at a key in Map
5.Dislplay by Iterator
6.Count Elements at a specific key
7.Exit
Enter your Choice: 1
Enter value to be inserted: 3
Enter the key: c

---------------------
Map Implementation in Stl

---------------------
1.Insert Element into the Map
2.Delete Element of the Map
3.Size of the Map
4.Find Element at a key in Map
5.Dislplay by Iterator
6.Count Elements at a specific key
7.Exit
Enter your Choice: 1
```

www.FirstRanker.com

www.FirstRanker.com

FirstRanker.com
Firstranker's choice
www.FirstRanker.com www.FirstRanker.com


```
Enter value to be inserted: 4
Enter the key: d

---------------------
Map Implementation in Stl

---------------------
1.Insert Element into the Map
2.Delete Element of the Map
3.Size of the Map
4.Find Element at a key in Map
5.Dislplay by Iterator
6.Count Elements at a specific key
7.Exit
Enter your Choice: 1
Enter value to be inserted: 5
Enter the key: e

---------------------
Map Implementation in Stl

---------------------
1.Insert Element into the Map
2.Delete Element of the Map
3.Size of the Map
4.Find Element at a key in Map
5.Dislplay by Iterator
6.Count Elements at a specific key
7.Exit
Enter your Choice: 3
Size of Map: 5

---------------------
Map Implementation in Stl

---------------------
1.Insert Element into the Map
2.Delete Element of the Map
3.Size of the Map
4.Find Element at a key in Map
5.Dislplay by Iterator
6.Count Elements at a specific key
7.Exit
Enter your Choice: 4
Enter the key at which value to be found: a
1
```

www.FirstRanker.com

www.FirstRanker.com

FirstRanker.com
Firstranker's choice
www.FirstRanker.com www.FirstRanker.com


```
----------
Map Implementation in Stl

----------
1.Insert Element into the Map
2.Delete Element of the Map
3.Size of the Map
4.Find Element at a key in Map
5.Dislplay by Iterator
6.Count Elements at a specific key
7.Exit
Enter your Choice: 5
Displaying Map by Iterator: a: 1
b: 2
c: 3
d: 4
e: 5

----------
Map Implementation in Stl

----------
1.Insert Element into the Map
2.Delete Element of the Map
3.Size of the Map
4.Find Element at a key in Map
5.Dislplay by Iterator
6.Count Elements at a specific key
7.Exit
Enter your Choice: 2
Enter the mapped string to be deleted: c

----------
Map Implementation in Stl

----------
1.Insert Element into the Map
2.Delete Element of the Map
3.Size of the Map
4.Find Element at a key in Map
5.Dislplay by Iterator
6.Count Elements at a specific key
7.Exit
Enter your Choice: 5
Displaying Map by Iterator: a: 1
```

www.FirstRanker.com

www.FirstRanker.com

FirstRanker.com
Firstranker's choice
www.FirstRanker.com www.FirstRanker.com

b: 2
d: 4
e: 5

Map Implementation in Stl

1.Insert Element into the Map
2.Delete Element of the Map
3.Size of the Map
4.Find Element at a key in Map
5.Dislplay by Iterator
6.Count Elements at a specific key
7.Exit
Enter your Choice: 7

www.FirstRanker.com

www.FirstRanker.com