Experiment 1

Q2

```
#include <iostream>
#include <string>
using namespace std;
class book {
  public:
    int price;
    int pages;
    string name;
    void accept() {
    cout << " Enter book name : ";
    cin >> name;
    cout << " enter price:";
    cin >> price;
    cout << " Enter pages: ";
    cin >> pages;
    }
      void display () {
       cout << " book name is : " << name;
       cout << " book price is: " << price;
       cout << " pages of book are : " << pages;
      }
       int get_price() {
       return price; }
    };
  int main() {
     book b1, b2;
     b1.accept();
     b2.accept();

if (b1.get_price() > b2.get_price()) {
cout << " book1 has greater price\n";
b1.display();
}
```

```
}
else {
cout << "book 2 has higher price \n";
b2.display();
}
return 0; }
```

* output

```
enter book name: programming_in_c
enter price   : 120
enter pages   : 230
enter book name: matilda
enter price   : 230
enter pages   : 120
book 2 has higher price
book name is: matilda
book price is : 230
pages of book one : 120
```

Q1

```
#include <iostream>
#include <string>
using namespace std;
class student {
public:
  int roll_no;
  string name;

  void display () {
  cout << "name is: " << name << endl;
  cout << "roll no is: " << roll_no << endl;
  }};
```

```
int main ()
{
  Student S1;
  S1. name = "Aayush";
  S1. rollno = 60;
  S1. display ();
  return 0;
  }
```

* output
name is Aayush
rollno is 60

Q3

```
# include <iostream>
  using namespace std;
  class time {
  public:
   int hours, mins, secs, totaltime;
   char col1, col2;
   void input ()
   {
    cout << " enter time in HH:MM:SS format: ";
    cin >> hours >> col1 >> mins >> col2 >> sec;
   }
   void calculate () {
  total time = (hours * 3600) + (mins * 60) + secs;
  }
   void display () {
    cout << " total time in seconds : " << total time;
  }};
```

```
int main () {
  time t;
  t. input ();
  t. calculate ();
  t. display (); }
```

* output

```
enter time in HH:MM:SS format:
2:30:15
total time in seconds : 9015
```

31/7/25

1

```
#include <iostream>
using namespace std;
class city {
  private:
    string name;
  public:
    int population;
    void accept() {
    cout << " enter name: ";
    cin >> name;
    cout << " enter population: ";
    cin >> population;
  }
  void display() {
    cout << "name: " << name << ", population" << population << endl;
  }};

int main() {
  city c[5];
    int i, max;
  for (i = 0; i < 5; i++) {
  c[i].accept();
  c[i].display();
}
max = c[0].population;
for (i = 1; i < 5; i++) {
  if (c[i].population > max) {
    max = c[i].population;
  }}
  cout << " max population is: " << max << endl;
  return 0; }
```

1) output

enter name : Pune
enter population 12000
name : pune , population : 12000
enter name : Mumbai
enter population : 1,50,000
name : mumbai, population : 150,000
enter name : Alibaug
enter population : 34000
name : alibaug, population : 34000
enter name : nashik
enter population : 10000
name : nashik , population : 10000
~~enter name :~~
~~enter population :~~
~~name :~~ ~~population :~~
Max population : 150000

2) #include <iostream>
using namespace std;
class Account {
private:
int Account_no, balance;
public:
void accept() {
cout << "enter amount no: ";
cin >> Account_no;
cout << "enter balance : ";
cin >> balance;
}

```
void check () {
  if (balance <= 5000) {
  balance += balance * 0.5; }
}
  void display () {
   if (balance <= 5250) {
   cout << " accountno: " << Account_no << endl
   cout << " balance: " << balance << endl; }
}
};
int main() {
 Account c[10];
 int i;
 for (i=0; i<10; i++) {
  cout << " enter details for account: " << i << endl;
  c[i].accept();
}
 for (i=0; i<10; i++) {
  c[i].check();
}
 for (i=0; i<10; i++) {
  c[i].display();
}
 return 0;
}
```

* output

enter details for account: 0
enter account no: 1
enter balance: 1200
-11- : 1
-11- : 2
-11- : 2000
-11- : 2
-11- : 3
-11- : 3400
-11- : 3
-11- : 4
-11- : 5000
-11- : 4
-11- : 5
-11- : 5000
-11- : 5
-11- : 6
-11- : 4000
-11- : 6
-11- : 7
-11- : 10,000
-11- : 7
-11- : 8
-11- : 6,000
-11- : 8
-11- : 9
-11- : 1000
-11- : 9
-11- : 10
-11- : 2300
-11-

account no : 1
account balance: 1260
-11- : 2
-11- : 2100
-11- : 3
-11- : 3570
-11- : 4
-11- : 5250
-11- : 5
-11- : 5250
-11- : 6
-11- : 4200
-11- : 9
-11- : 1050

c)

```
#include <iostream>
#include <string>
using namespace std;
class staff {
  private:
  string name post;
  public:
  void accept() {
  cout << "enter name: ";
  cin >> name;
  cout << "enter post: ";
  cin >> post;
}
Void check() {
    if (post=="Hop" || post=="head") {
    cout << "head staff name : " name << endl;
}
};
```

```
int main () {
  staff c[5];
  int i;
  for (i=0; i<5; i++) {
  c[i].accept();
}
  for (i=0; i<5; i++)
    c[i].check();
  } return 0;
}
```

* output

```
enter name: Mrunal_Awane
enter post: HOD
      name: Pallavi_Nehete
      post: HOD
      name: Vilas_Pathod
      post: lecturer
      name: Meghna_Dhotre
      post: teacher
      name: Shilpa_Budhavale
      post: HOD
headstaff name : Mrunal_Awane
               : Pallavi_Nehete
               : Shilpa_Budhavale
```

3117

Experiment 3

1)
```
#include <iostream>
using namespace std;
class Book
{
int price;
string book_title;
string author_name;
public:
void accept()
{
cout << "Enter book title, author name and price
      of your book: ";
(in >> book_title >> author_name >> price;
}
void display()
{
cout << "In Book_title is: " << book_title;
cout << "In Name is: " << author_name;
cout << "In price is: " << price;
}
};
int main()
{
Book b1;
Book *p = &b1;
p -> accept();

return 0;
}
```

Output

```
Enter book title, author name and price
of your book: Any man
Any main
450
Book title is: Any man
Name is: Any Man
price is: 450
```

2)
```
#include <iostream>
using namespace std;
class student
{
public
int roll;
string name;
float perc;
void accept()
{
cout << "Enter roll no: " << endl;
cin >> this -> roll;
cout << "Enter name: " << endl;
cin >> this -> name;
cout << "Enter percentage: " << endl;
cin >> this -> perc;
}
void display
{
this -> accept()
{
this -> accept()
cout << "roll no of student" << this -> roll << endl;
cout << "name of student" << this -> name << endl;
cout << "percentage of student" this -> perc << endl;
}
};
int main() {
student s;
s.display();
}
```

3) #include <iostream>
using namespace std;

class student
{
public:
int roll;
string name;
void accept()
{
cout << "enter roll no & name: ";
cin >> roll >> name;
}
class marks
{ public
int cpp_marks;
int c_marks;
float percentage;
int add;
float d;
void accept()
{
cout << " enter c & c++ marks: ";
cin >> c_marks >> cpp_marks;
}
void display
{
cout << "\n c marks are: " << c_marks;
cout << " cpp marks are: " << cpp_marks;
add = c_marks + cpp_marks;
d = (float) add / 200;

```
percentage = d*100;
cout << "In percentage is :" <<percentage << "%";
}
};
};
int main (){
Student S1;
S1.accept();
Student:: marks m1;
m1.accept();
m1.display();
return 0;
}
```

output

Enter rollno & name of Student: 17
pooja
Enter C& CPP marks: 95
89

C marks : 95
CPP marks : 89
percentage is : 92%

13/8

Experiment-4

1)
```
#include <iostream>
using namespace std;
class Number {
int num;
public:
  void setNum(int n)
{ num = n; }
void swap(number & obj)
{ int temp = num;
  num = obj.num;
  obj.num = temp; }
void display()
{ cout << "Number: " << num << endl; } };
int main()
{ Number n1, n2;
  n1.setNum(10);
  n2.setNum(20);
cout << "Before swap: " << endl;
n1.display();
n2.display();
n1.swap(n2);
cout << "After swap: " << endl;
n1.display();
n2.display();
```

Output:
Before Swap:
Number: 10
Number: 20
After Swap:
Number: 20
Number: 10

2) #

```
#include <iostream>
using namespace std;
class Number {
int num;
public:
void setnum(int n)
{ num = n; }
void display()
{ cout << "Number: " << num << endl; }
friend void swap (Number &n1, Number &n2); }
void Swap(Number &n1, Number &n2); {
int temp = n1.num;
n1.num = n2.num;
n2.num = temp;
int main()
{ Number n1, n2;
n1.setnum(10);
n1.setnum(20);
cout << " Before Swap: " << endl;
n1.display();
n2.display();

Swap (n1, n2);
cout << " After Swap: " << endl;
n1.display();
n2.display();
return 0; }
```

output

Before swap:
Number: 10
Number 20
After swap
Number 20
Number 10

3)
```
#include <iostream>
using namespace std;
class Number 2;
class Number 1
{ int num;
public :
  void set Num (int n)
{ num = n; }
void display()
{ cout<< "Number 1: " << num << endl; }
  friend void swap (Number 1 &n1, Number 2 &n2); }

class Number 2
{ int num;
public:
void set Num (int n)
{ num = n; }
void display ()
{ cout << "Number 2: " << num << endl; }
friend void swap (Number 1 &n1, Number 2 &n2); }

Void Swap (Number 1 & n1, Number 2 &n2)
{
int temp = n1.num;
n1. num = n2. num;
n2. num = temp; }
```

output

```
Before Swap :
Number 1 : 10
Number 2 : 20
Number
After swap :
Number 1 : 20
Number 2 : 10
```

4)

```
#include <iostream>
using namespace std;
class n2;
class n1 {
  int a;
  public:
  void accept(){
  cout << "Enter first number: ";
  cin >> a;
}
class n2 {
    int b;
    public : void accept(){
    cout << "Enter second number: ";
    cin >> b;
  }
    friend void greated(n1 &s, n2 &r);
  };
void greates(n1 &s, n2 &r){
  float g = s.a;
  if(s.a > r.b) {
  cout << " greater number is " << s.a;
  else {
    cout << " Greater number is: " << r.b;
int main() {
  n1 s;
  n2 r;
  s.accept();
  r.accept();
  greatest(s,r);
```

6)

5)
```
#include<iostream>
using namespace std;
class demo
{
public:
    int p, q;
    void accept()
    {
    cout<<" Enter 2 nos:-"<<endl;
    cin>>p>>q;
    }
    void display()
    {
        cout<<" After swapping:"<<" value of p="
        <<p<<" value of q="<<q;
    }
    void swap(demo &t)
    {
    int temp=t.p;
    t.p=t.q;
    t.q=temp;
};
int main()
{
    demo k;
    k.accept();
    k.swap(k);
    k.display();
}
```

```
#include <iostream>
using namespace std;
class B;
class A {
    int a;
public:
    void accept1() {
    cout << "Enter first no." << endl;
    cin >> a;
    }
    friend int sum(A p, B q);
};
class B {
    int b;
public:
    void accept2() {
    cout << "Enter the 2 no " << endl;
    cin >> b;
    }
    friend int sum(A p, B q);
};
int sum(A p, B q) {
    int sum;
    sum = p.a + q.b;
    return sum;
}
int main() {
    A k;
    B f;
    k.accept1();
    f.accept2();
    cout << "The sum of 2 nos. " << sum(k, f);
}
```

7 #

```
using ...
class A {
    int a, b;
public:
    void accept () {
        cout << " Enter 2 nos: " << endl;
        cin >> a >> b;
    } friend void swap numbers (A &t);

    void display () {
        cout << " Value of a : " << a;
        cout << " value of b : " << b;
    } friend void swap numbers (A &t); };
void swap numbers (A & t) {
    int temp = t . a;
    t . a = t. b;
    t . b = temp; }
    int main () {
        A k;
        k . accept ();
        swap numbers (k);
    k . display (); }
```

#..
using ---
class cube;
class box {
    int l, b, h, v1;
public:
    void accept() {
        cout << "Enter dimension of box" << endl;
        cin >> l >> b >> h;
    }
    friend void greater volume (box p, cube q);
};
class cube {
    int side, v2;
public:
    void accept2() {
        cout << "Enter the dimensions of cube:" << endl;
        cin >> side; }
friend void greater volume (box p, cube q); };
void greatervolume (box p, cube q) {
    p.v1 = p.l * p.h * p.b;
    q.v2 = q.side * q.side * q.side;
    if (p.v1 > q.v2) {
        cout << "the box having greater volume:" << endl;
    } else {
        cout << "the box having greater vol is" << endl;
    }}
int main () {
    box k;
    cube f;
    k.accept1();
    f.accept2();
    greatervolume(k, f);
}

14/11

Exp 5

a)

```
#include <iostream>
using namespace std;
class sum {
int n;
int sum;
public:
  sum (int num) {
    n = num;
    sum = 0;
    for (int i=1; i<=n; i++) {
      sum = sum + i;
  } }
    void display () {
      cout << " sum of nos: " << sum << endl; }
int main() {
  int n;
  cout << " Enter value of n: ";
  cin >> n;
  sum s(n);
  s.display(); }
```

b) #
us
cl

```
#. . . . . .
using . . . . . .
class student {
  string name;
  float perc;
public :
    student (string n, float p) {
    name = n;
    percentage = p;
  }
  void display () {
  cout << " Name: " << name << "\n percentage: " << perc;
  }};
int main () {
  string name;
  float perc;
  cout << " Enter name: ";
  cin >> name;
  cout << " Enter percentage : ";
   cin >> perc;
  Student s1 (name, percentage);
  cout << " student Details: ";
  s1. display ();
}
```

C) #. . . . . .

using . . . . . ;

class college {

int roll_no ;

string name, course;

public:

college (int r, string n) {

roll_no = r ;

name = n ;

course = " computer Engineering " ;

}

void display () {

cout << "\n Roll No : " << roll_no << endl;

cout << " Name: " << name << endl;

cout << " course: " << course << endl;

} };

int main () {

college s1 (5, " Ayush ");

college s2 (6, " Shone ");

s1. display ();

s2. display ();

}

```
. . . . . .
using . . . std ;
class college {
int roll_no ;
string name, course ;
public :
college () {
        roll_no = 601
        name = "Aayush" ;
        course = " Computer engineering" ;
    }
college (int r, string n, string c) {
    roll_no = r ;
    name = n ;
    course = c ; }
void display () {
cout << " Roll No : " << roll_no << endl ;
cout << " Name : " << name << endl ;
cout << " course : " << course << endl ; } } ;
int main () {
    college s ;
    college s1 (s, " Aarav ", " CSE ") ;
  s. display () ;
  s1 . display () ;
}
```

14/11

Exp 6

1)
```
#include <iostream>
using namespace std;
class person {
    protected:
    string name;
    int age;
};
class student : protected person {
    private:
    int roll;
    public:
    void accept () {
    cout << " Enter Name: ";
    cin >> name;
    cout << " Enter Age: ";
    cin >> age;
    cout << " Enter Roll no: ";
    cin >> roll; }
    void display () {
    cout << " Name and Age: " << name << age;
    cout << " Roll no: " << roll; };
int main () {
    Student s;
    s. accept ();
    s. display ();
}
```

2)

2) #.....
using .....
class Academic {
protected:
int marks; };
class sports {
protected:
int sp-score;
};
class Result: protected Academic, protected sports {
private:
float pv;
public:
void accept() {
cout<<"Enter marks in academics & sports: ";
cin >> marks >> sp-score; }
void calculate() {
int total = marks + sp-score;
per = (total / 200.00) * 100;
cout << "Result = " << per << endl;
}}
int main() {
Result r;
r.accept();
r.calculate();
}

3) #.....

using .....

```
class vehicle {
    public:
    string brand, model;
};
class car: public vehicle {
    protected:
    string type; };
class ecar: protecter car {
    private:
    int battery cap;
    public:
    void accept() {
    cout << "Enter brand Name: ";
    cin >> brand;
    cout << "Enter model ";
    cin >> model;
    cout << "Enter type of car";
    cin >> type;
    cout << "Enter battery capacity:";
    cin >> battery cap; }
void display() {
    cout << "Brand and model:" << brand << model;
    cout << "type of car :" << type << endl;
    cout << "Battery capacity." << battery cap << endl;
};
int main() {
ecar e;
e.accept();
e.display();
}
```

4) #.....

```
using name ..... std;
class Employee {
    protected:
        int emp_id;
        string name;
};
class manager : public Employee {
    private:
    string dept;
    public:
    void accept() {
    cout << "Enter Employee ID: ";
    cin >> emp_id;
    cout << "Enter name of employee: ";
    cin >> name;
    cout << "Enter department: ";
    cin >> dept; }
    void display() {
    cout << " Employee ID & Name: " << emp_id <<
    name;
    cout << "Department: " << dept << endl;
} };
class Developer: protected Employee {
    private:
    string prog_lang;
    public:
    void acc() {
    cout << " Enter programming language: ";
    cin >> prog_lang; }
    void disp() {
    cout << " programming language: ";
    cin >> prog_lang; }
```

```
void disp() {
 cout << " programming language : " << programming
} };
int main() {
    manager m;
    m.accept();
    m.display();
    Developer d;
    d.acc();
    d.disp();
}
```

14/11

Exp 7

a)

```
#include <iostream>
using ..... ;
class area1 {
    public:
    int l, b;
    void area(int a, int b) {
    int c = a * b;
    cout << " Lab Area : " << c << "Sq "; }
    void area(int s) {
    int A = s*s;
    cout << " class Area : " << f << "Sq "; };
int main() {
    area1 m;
    m.area(20, 30);
    cout << endl;
    m.area(20);
}
```

6) #include <iostream>
using namespace std;
class sum1 {
public:
int i;
void sum(float a[5]) {
float s = 0;
for (i = 0; i<5; i++) {
s + a[i]; }
cout<<"sum of 5 float nos: "<<s<<endl; }
void sum(int b[10]) {
int s = 0;
for (i=0; i<10; i++) {
s += b[i]; }
cout<<" sum of 10 integer nos: "<<s<<endl; }};
int main() {
sum s1;
float[5];
int d[10];
cout<<"Enter 5 float Nos: \n";
for (int i=0; i<5; i++) {
cin >> c[i]; }
cout<<"Enter 10 int nos: \n";
for (int i=0; i<10; i++) {
cin >> d[i]; }
s1.sum(c); s1.sum(d); }

c #. . . . . .

using ----.

class num {

int a;

public:

void accept() {

cout << "Enter value of a: ";

cin >> a;

}

void displ () {

cout << " value of a: " << a; }

void operator - () {

a = - a; } };

int main () {

num n1;

n. accept();

-n1;

n1 . displ (); }

---

d) # . - — — — .

using - — — — —

Class num {

int a, b, c;

public:

void accept () {

cout << "Enter val of a: ";

cin >> a >> b; }

void disp () {

cout << " value of a: " << a; }

void operator ++ () {

a = ++ a; } };

int main () {

num n1;

n1 . accept (); ++n1; n1. disp (); }

14/11

Exp 8

```
#include . . _ _ _ _
#include <string>
using . . _ _ _ _ _ _
class abc {
public:
string str;
void acc() {
    cout << "Enter string: ";
    cin >> str; }
abc operator + (abc c) {
abc temp;
temp.str = str + s.str;
return temp; }
Void disp() {
    cout << "concatenated string: " << str << endl; };
int main () {
abc s1, s2, r;
s1 . acc();
s2 . acc();
r = s1 + s2;
r . disp();
return 0; }
```

o/p

Enter string: Hello
Enter string: world
Concatenate string: Hello World

2) #. . . . . .
using . . . —
class ILogin {
protected
string name, password;
public:
void accept() {
cout << "Name: ";
cin >> name;
cout << "password: ";
cin >> password; }};
class EmailLogin : virtual public ILogin {
public:
void showEmail() {
cout << Name << " " << password << endl;};
class membershiplogin : virtual public ILogin {
public:
void showmembership() {
cout << name << " " << password << endl;};
Class Employee: public EmailLogin, public membership {
public:
Void input() {
accept(); }
void display() {
showEmail();
showMembership() };
int main() {
Employee e;
e.input();
e.display();
return 0;
}

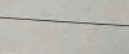
14/11

Exp 9

```
#include <iostream>
#include <fstream>
#include <string>
#include <string>
using
int main () {
    ifstream fin;
    ofstream fout;
    fin.open ("source.txt");
    fout.open ("destination.txt");

    if (!fin) {
        cout << "error takes place" << endl;
        return 1; }
    char ch;
    while (fin.get(ch)) {
        fout.put (ch);
    }
    fin.close();
    fout.close();
    fin.open ("source.txt");
    string word;
    int wordcount = 0;
    while (fin >> word) {
        wordcount ++;
    }
    cout << "The wordcount is:" << wordcount << endl;
    fin.close();
    fin.open ("source.txt");
    string target;
    int count=0;
    cout << " Enter the target " << endl;
```

```
cin >> target
while (fin > word) {
    if (word == target) {
     count ++; }}
cout << " the target word found whose occurance is "
count << endl;
fin.open (" source.txt ");
int digit count = 0;
int space count = 0;
while (fin.get (ch)) {
    if (isdigid (ch)) {
        digit count ++;
    }
    if (isspace (ch)) {
        space count ++; }}
cout << " digit count is: " << digit count;
cout << " the space count is: " << space count;
fin.close(); }
```



Exp 10

```
using . . . . .
template <class T>
T sum (T a[], int n) {
    T Sum = 0;
for (int i = 0; i<n; i++) {
    Sum += a[i];
}
return Sum; }
int main() {
    int n = 5;
    int int a[n]
    double doublea[n];
cout << " Enter 5 integer numbers ",
for (int i=0; i<n; i++) {
cin >> int a[i]; }
cout << " Enter 5 integer elements: ",
for (int i = 0; i<n; i++) {
    cin float a[i];
}
cout << " Enter 5 double numbers: ";
for (int i=0; i<n; i++) {
    cin >> doublea[i]; }
cout << " Sum of integer numbers = " << sum (int a, n);
cout << " Sum of foat numbers = " << sum (float a, n);
cout << " Sum of doubleelements = " << sum (double a, n);
}
```

b)
```
#include<string>
using namespace std;
template<class T>
T square(T x){
    return x*x;
}
template<>
string square<string>(string s){
    return s+s;
}
int main(){
    int num;
    string str;
    cout<<"Enter an integer: ";
    cin>>num;
    cout<<"Enter a string: ";
    cin>>str;
cout<<"\n Square of integer "<<num<<"="<<square(num);
cout<<"\n Square of string "<<str<<"="<<square(str);
}
```

```
using:- - - - -
template <<class T>
class calc {
public:
  T a, b;
  void accept() {
  cout << "Enter 2 nos: ";
   cin >> a >> b; }
  void add() {
  cout<<" Addition is" << a+b << endl; }
  void sub() {
  cout << "Subtraction is" << a-b << endl;
  }
  void mul() {
  cout << " multiplication is: " << a*b << endl;
  }
  void div() {
  cout << " Division is: " << a/b << endl;
  };
int main() {
cal <double> n;
int ch;
n.accept();
while(1) {
cout<<"1. Addition \n 2. Subtraction \n 3. multiplication
\n 4. Division \n 5. Exit";
Cout << "Enter choice:";
cin >> ch;
Switch (ch) {
   case 1: n.add(); break;
```

```
case 2 : n.sub() ; break;
case 3 : n.mul() ; break;
case 4 : n.div() ; break;
case 5 : exit(0) ; break;
default << "Wrong choice: ";
    break;
  }
 }
}
```

Exp 11

```
#include <iostream>
#include <vector>
#include <ctype>
using namespace std;
int main() {
  vector <int> vec [5];
  int i;
  cout << "Enter 5 vector elements: ";
  for (i = 0 ; i <5; i++) {
    cout << vec [i] >> endl;
  }
  cout << " Modified elements: ";
  for (i = 0; i <5 ; i++) {
  vec [i] = vec[i] + 1 * 2;
  }
  for (i = 0; i <5 ; i++) {
     cout << vec [i] << " "; }
  cout << endl;
  int scalar;
  cout << " Enter a scalar val to multiply: ";
  cin >> scalar;
  cout << "Enter 5 Scalar val After multiplying: ";
  for (i = 0; i <5 ; i++) {
  vec [i] = vec[i] * scalar)
  for (i=0; i<5; i++) {
  cout << vec [i] << " "; }
  cout << endl;
}
```

Qn

14/11

Exp 12

1) #include <iostream>
#include <stack>
#include <ctype>
using - - - -
int main() {
stack <int> V;
V.push(1);
V.push(2);
V.push(3);
V.push(4);
V.push(5);

if (V.empty()) {
cout << "stack is not empty"; }
else {
cout << "In stack is not empty"; }
cout << "\n size: " << v.size();
cout << "In topmost;" V.top();

cout << "In Stack: ";
while (!V.empty())
{ cout << v.top() << " "; V.pop; }
cout << "In Size after popping: " << v.size();
}

```
#include <queue>
#include <stype>
using namespace std;
int main() {
queue <int> v;
v.push(11);
v.push(22);
v.push(33);
v.push(44);
v.push(55);
if (v.empty()) {
cout << "\n queue empty"; }
else {
cout << "\n queue is not empty"; }
cout << "\n size: " << v.size();
cout << "\n Front: " << v.front();
cout << "\n Back: " << v.back();
cout << "\n queue: ";
while (!v.empty()) {
cout << v.front() << " "; v.pop(); }
cout << "\n size after popping: " << v.size(); }
```

14/11