Experiment 1

a)
```
#Include <iostream>
using username std;
class student {

 private:
  string name;
  int roll;
  int cls;

 public:
  void accept (){
     cout << "Enter the name of student:";
     cin << name;
     cout << "Enter roll no.:";
     cin >> roll;
     cout << "Enter the class:";
     cin >> cls;
    }
  void ~~dist~~ display (){
     cout << "Student details:";
     cout << "\nStudent name:" << name << endl;
     cout << "Roll no.:" << roll << endl;
     cout << "Class:" << cls << endl;
    }
};

int main (){
  student s1;
   s1.accept();
   s1.display();
   return 0;
}
```

Output

```
Enter the name of student : Sam
Enter roll no. : 03
Enter the class : 10
Student detail :
Student details : Sam
Roll no. : 03
Class : 10
```

b) #include <~~stdio~~ <iostream>
using namespace std;
class book {
private:
string Book;
int page;

public:
float price;
void accept(){
cout << "Enter book name:";
cin >> name;
cout << "Enter price:";
cin >> price;
cout << "Enter pages:";
cin >> page;
}
void display(){
cout << "In Book details:";
cout << "In Book name:" << name;
cout << "In Price:" << ~~pag~~ price;
cout << "In ~~pa~~ Pages:" << page;
}
};

int main()
{
book b1, b2;
b1.accept();
b2.accept();

```
if(b1.price > b2.price)
 {
    b1.display();
 }else{
       b2.display();
       }
 return 0;
}
```

Output.

```
Enter book name: ABC
Enter price: 50
Enter pages: 50
Enter book name: abc
Enter price: 100
Enter pages: 100

Book details:
Book name: abc
Price: 100
Pages: 100
```

c) #include <iostream>
using namespace std;
class time {

public:
int h, m, s, ts;
char col1, col2;

void input(){
cout<<"Enter time in HH:MM:SS format:";
cin>>h>>col1>>m>>col2>>s;
}
void cal(){
ts = (h*60*60)+(m*60)+s;
}
void display(){
cout<<"Total time in seconds:"<<ts;
}
};
int main(){
time t1;
t1.input();
t1.cal();
t1.display();
}

Output.

Enter time in HH:MM:SS format: 01:01:60
Total time in seconds: 3720

Or

16/07/2025

Experiment 2

a)
```
#include <iostream>
using namespace std;
class city {
    public:
        string name[5];
        int pop[5];
        int max;
        int i;
        int k;
    void accept(){
        cout<<"Enter city name and population:";
        for(i=0; i<5; i++)
        {
            cin>>name[i]>>pop[i];
        }
    }
    void display(){
        max=pop[0]
        for(i=0;i<5;i++){
            if(pop[i]>max)
            {
                max=pop[i];
                k=i;
            }
        }
cout<<"City with greater population: "<<name[k]<<"
    }
} c;

int main(){
    c.accept();
    c.display();
    return 0; }
```

Output

Enter the name of city and population:
a 100
b 50
c 200
d 150
e 300
City with greater population: e 300.

16/02/2025

b)
```
#include <iostream>
using namespace std;
class acc{
        public:
         int acc_no[3];
         float bal[3];
         int i;
        float balance;
    void accept(){
    cout<<"Enter the account number and balance :";
    for(i=0; i<3; i++){
    {
      cin>> acc_no[i] >> bal[i];
    }
  }
void display ()
    {
     for(i=0; i<3;i++)
     {
       if(bal[i]>=5000)
     {
       balance = bal[i]+(bal[i]* 10* 1)/100;
   cout<<"Account name and balance after 10% increase:"
       << acc_no[i] <<"  " << balance <<endl;
      }
    }
   }
 } c;
int main(){
   c.accept();
   c.display();
   return 0;
}
```

Output

Enter the account number and balance: 1 100000
2 5000 500
3 6000

Account name and balance after 10% increase : 1 110000
Account name and balance after 10% increase: 3 6600

16/07/2025

c)
```
#include <iostream>
using namespace std;
class staff {
    public:
     string name[3];
     string p[3];
     int i,j;
  void accept(){
  cout<<"Enter the name and post:";
   for(i=0;i<3;i++)
    {
      cin>>name[i]>>p[i];
    }
  }
void display()
  {
   for(i=0;i<3;i++)
   {
     if(p[i]=="HOD")
   {
   cout<<"Names of staff who are HOD:"
            <<name[i]<<endl;
   }
  }
  }
};
int main()
{
  C.accept();
     c.display();
     return 0;
  }
```

Output

```
Enter name and post : a HOD
b lecturer
c HOD
Name of staff who are HOD: a
Name of staff who are HOD: c
```

30/7/25

16 | C Experiment 3 [Pointer to object]

a)
```
#include <iostream>
using namespace std;
class book {
        string title;
        string author_name;
        float price;

    void accept(){
cout<<"Enter book title, author name and price:";
    cin>>*title>>author_name>>price;
        }
    void display(){
cout<<"Book title, author name & price : "<<title<<"
    author_name<<" "<<price<<endl;
    }
} b1;

int main(){
        book *p;
        p=&b1;
    p->accept();
    p->display();
    }
```

[This Pointer]

b)
```
#include <iostream>
using namespace std;
class book {
    public:
        int roll;
        float percentage;

    void accept (int roll, float percentage) {
        this -> roll = roll;
        this -> percentage = percentage;
    }
void display (){
    cout << "Roll number and percentage: " << this -> roll << " "
        this -> percentage << endl;
    }
}b1;

int main(){
    b1.accept(7,25);
    b1.display();
    return 0;
}
```

Output

Roll number and Percentage: 7 25

[Nested Class]

c) #include <iostream>
using namespace std;
class studentmarks {
public:

class student {
public:
int roll;
float percentage;

void accept() {
cout << "Enter roll number and percentage: ";
cin >> roll >> percentage;
}
void display() {
cout << "Roll number and percentage : " << roll << "
percentage << endl;
}
};
};
int main() {
studentmarks:: student s1;
s1.accept();
s1.display();
return 0;
}

Output.
Enter roll number and percentage: 2 55
Roll number and percentage: 2 55

30/7/25

Exp 4

1)
```
#include <iostream>
using namespace std;
class Num{
        int a;

public:
    void Num (int v){
            a = v;
                }
void display(){
   cout << value << endl;
  }
void swap(Num &obj){
      int temp = a;
            a = obj.a;
          obj.a = temp;
        }
  };
int main(){
     Num  n1(10), n2(20);
cout << "Before swap: " << endl;
cout <<"num1 = " << n1.display;
cout << "num2= " << n2.display;

  n1.swap(n2);

Cout << "In After swap: " << endl;
cout << "num1=" << n1.display();
cout << "num2 = " << n2.display();

return 0;
}
```

2) #include <iostream>
using namespace std;
class Num {
    int value;
public:
    ~~Num(int i)~~ void accept(int v)
    { v = value; }

friend void swap(Num&, Num&); void show(){
    cout << value << endl;
    }
};

void swap1(Num &a, Num &b){
        int temp = a.value
        ~~a~~ = a.value = b.value;
        b.value = temp;
    }
int main(){
    Num n1, n2;
    n1.accept(5); n2.accept(10);
    cout << "Before swapping: \n";
    n1.show(); n2.show();
    swap1(n1, n2);
    cout << "After swapping: \n";
    n1.show(); n2.show();
    }

3) #include <iostream>
using namespace std;
class B;
class A {
    int a;
  public:
    void accept(){
    void accept (int x)
    { a = x; }
};
friend void swap (A obj1, B obj2);
};
class B {
    int b;
  public:
    void accept (int y)
    { b = y; }

friend void swap (A obj1, B obj2);
};
void swap (A obj1, B obj2)
{
    int temp = obj1.a;
        obj1.a = obj2.b;
        obj2.b = temp;
cout<<" values after swapping: " <<obj1.a<<obj2.b;
}
int main (){
    A obj1;
    B obj2;
    obj1.accept(); obj2.accept();
    swap (obj1, obj2);
    return 0;

4)
```
#include<iostream>
using namespace std;
class Result2;
class Result1{
  public:
    int o;
  void accept()
  { cout<<"Enter the value of o: ";
    cin>>o;
    }
  friend void avg(Result1 r1, Result2 r2);
  }; r1;
class Result2{
    public:
     int i;
   void accept()
   { cout<<"Enter the value of i: ";
     cin>>i;
     }
friend void avg(Result1 r1, Result2 r2);
 }; r2;
void avg(Result1 r1, Result2 r2)
   { float avg;
      avg=(r1.o+r2.i)/2;
  cout<<"Average of two results is: "; <<avg;
  }
int main(){
     r1.accept();
    r2.accept();
     avg(r1,r2);
  return 0;
  }
```

5) #include <iostream>
using namespace std;
class B;
class A{
public:
int o;
void accept(){
cout<<"Enter first number: ";
cin >> o;
}
friend void comp(A a1, B b1);
} a1;
class B{
public:
int i;
void accept(){
cout <<"Enter second number: ";
cin>>i;
}
friend void comp(A a1, B b1);
} b1;
void comp(A a1, B b1){
if(a1.o > b1.i)
{ cout<<"Greatest number is: "<<a1.o;}
else{
cout<<"Greatest number is: "<<b1.i;}
int main(){
a1.accept();
b1.accept();
comp(a1, b1);
return 0;
}

9/13/8

Experiment 5

(Parameterized)

a)

→

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

class Num
{   int sum=0;
    int n;

  public:
    int i;
   Num(int x){
      n=x;
   for(i=0;i<=n;i++)
    {
       sum = sum+i;
    }
   }
void disp(){
    cout<<" Sum =  "<<sum;
  }
};
int main(){
    Num n1(5);
    n1.disp();
     return 0;
}
```

b)

→

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

class Student{
    string name;
     float per;
  public:

  Student (string n, float p)
     {
       name = n;
       per = p;
     }
  void disp(){
    cout<<"Name and Percentage: "<<name<<"
     }
  };
int main(){
    Student s1("Sam", 88.8);
     s1.disp();;
     return 0;
  }
```

Q

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

class College {
        string name;
        int roll;
        string course;

  public:

College(string n, int r, string c="Computer Science")
      {
          name = n;
          roll = r;
          course = c;
      }
void disp(){
cout<<"Name, Course & Roll no.:"<<name<<" "<<
    course<<" "<<roll<<endl;
    }
};
int main(){

  College s1("Sam", 3);
  College s2("Shreyas", 75);
   s1.disp();
   s2.disp();

  return 0;
}
```

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

class Box {
    int length;
    int width;
public:
    Box() {
        length = 1;
        width = 5;
    }
    Box(int l, int w) {
        length = l;
        width = w;
    }
    void disp() {
        cout << "Length & width: " << length << " " << width;
    }
};

int main() {
    Box b1;
    b1.disp();

    Box b2(5, 3);
    b2.disp();

    return 0;
}
```

Experiment 6

1) Single Inheritance

→

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

class person{
    protected:
        string name;
        int age;
};
class student: protected person{
        int roll;
    public:
     void accept(){
        cout<<"Enter name, age and roll: ";
        cin >> name >> age >> roll;
      }
  void dis(){
cout<<" Name, age and roll number: " << name <<" " <<age
        " " << roll << endl;
    }
};
int main(){
   student s1;
   s1.accept();
   s1.dis();
}
```

3) Multiple Inheritance

→
```
#include <iostream>
using namespace std;
class academic {
  protected:
    int m1, m2;
};
class sport{
  protected:
    int score;
};
class result : protected academic , protected sport {
  public:
    void accept(){
cout <<"Enter academic marks and sports score: ";
cin >> m1 >> m2 >> score;
  }
  void cal(){
    int total;
    total = m1 + m2;
cout <<"Academic total marks and sports score: " << total <<"
    << score << endl;
  }
};
int main(){
  result r1;
  r1.accept();
  r1.cal();
}
```

3) Multilevel Inheritance

→
```
#include <iostream>
using namespace std;
class vehicle{
    public:
    string brand;
    string model;
};
class car : protected vehicle{
    protected:
    string type;
};
class Elecar : protected car{
        int batcap;
    public:
void acc(){
cout<<" Enter brand ,model , vehicle type & battery capacity :
cin >> brand >> model >> type >> batcap;
}
void dis(){
cout << " Brand ,model ,vehicle type ,& battery capacity : "<<
brand << model << type << batcap << endl;
}
};
int main(){
    Elecar e1;
    e1.acc();
    e1.dis();
}
```

4) Hierarchical Inheritance

→

```
#include <iostream>
using namespace std;
class emp {
   public:
   int id;
   string name;
  };
class manager: protected emp {
   string dept;
  public:
   void acc(){
cout<<"Enter employee id, name and department name:"
cin >> id >> name >> dept;
  }
void dis(){
cout<<"Employee name, id and department:" << name << " "
id<<" " << dept << endl;
   }
};
class developer: protected emp {
    string proglang;
  public:
    void acce(){
cout << "Enter employee name and programming language:";
cin >> name >> id >> proglang;
   }
void dis(){
cout<<"Employee name & programming language:" << name<<"
proglang << endl;
 } };
```

```
int main(){

    manager m1;
    developer d1;

    m1.acc();        m1.acc();
    d1               d2.acce();
    m1.dis();        m2.dis();
                     d2.dis();
return 0;   }
```

5) Hybrid Inheritance.

→
```
#include <iostream>
using namespace std;
class person{
  protected:
    string name;
    int age;
};
class academic{
  protected:
    int m1, m2;
};
class sport{
  protected:
    int score;
};
class student : protected person{
    int roll;
  public:
void accept(){
cout<<"Enter name, age and roll number: ";
cin>>name>>age>>roll;
}
void display(){
cout<<"Name, age and roll number of the students: "<<name<<
" "<<age<<" "<<roll<<endl;
}
};
```

```
class result : protected academic , protected sport {
  public:
    void accept() {
   cout <<"Enter academic marks and sports score: ";
   cin>> m1 >> m2 >> score;
   }
  void cal() {
      int total;
      total = m1 + m2;
 cout <<"Academic total marks and sports score: " <<
 total <<" " << score << endl;
    }
 };

int main(){
   student s1;
    s1.accept();
    s1. display ();
  result r1;
  r1. accept();
  r1. cal();
}
```

6) Virtual ~~Class~~ Base Class

→
```
#include <iostream>
using namespace std;
class college{
    public:
     int id;
    void accept1(){
     cout<<"Enter ID: ";
     cin>>id;
     }
};
class test:public virtual college{
      public:
       float p;
      void accept2(){
        cout<<"Enter Percentage: ";
        cin>>p;
        }
    };
class sports:public virtual college{
    public:
       int grade;
      void accept3(){
       cout<<"Enter grade: ";
       cin>>grade;
      }};
```

```
class result: public test , public sports {
        public:
          void print(){
              accept1();
              accept2();
              accept3();
      cout << "ID: " << id << endl;
      cout << "Percentage: " << p << endl;
      cout << "Grade: " << grade;
       } };
int main(){
    result r;
    r.print();
}
```

Q1

24/9/25

Experiment 7

a)

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

class Area {
  public:
    int area (int side){
        return side * side;
      }
    int area (int l, int w){
        return l * w;
      }
  };
int main (){
    Area c;
    int side, l, w;

cout << "Enter side of classroom (square): ";
cin >> side;
cout << "Area of classroom: " << c.area (side) << en

cout << "Enter length and width of laboratory (rectangl
cin >> l >> w;
cout << "Area of laboratory: " << c.area (l, w) <<

return 0;
}
```

b)

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

class sum {
public:
    float sum(float arr[], int n){
        float total = 0;
      for(int i = 0 ; i<n ; i++)
        total += arr[i];
        return total;
    }
  int sum(int arr[], int n){
        int total=0;
        for(int i=0 ; i<n ; i++)
          total += arr[i];
        return total;
    }
  };
int main(){
   sum c;

   float f[5] = {1.1, 2.2, 3.3, 4.4, 5.5};
   int i[10] = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10};

cout<<"Sum of 5 float values: "<<c.sum(f,5)<<end
cout<<"Sum of 10 int values: "<<c.sum(i,10)<<end

  return 0;
}
```

c)

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

class number {
    int a;
  public:
    void accept(){
      cout << "Enter val of a : " ;
      cin >>a ;
    }
   void display(){
     cout << "Value of a : " <<a ;
   }
   void operator -(){
       a = -a;
     }
  };
int main(){
    number n1;
   n1.accept();
   -n1;
   n1.display();
}
```

d)

→

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

class num{
    int a, b;
public:
  void accept(){
   cout<<"Enter value of a and b: ";
   cin>>a>>b;
  }
  void display(){
   cout<<"value of a & b: "<<a<<" "<<b<<endl;
  }
void operator ++(){
      ++a;
  }
void operator ++(int){
       b++;
    }
};
int main(){
  number n1, n2;

  n1.accept();
  ++n1;
  n1.display();

  n2.accept();
  n2++;
  n2.display();

  return 0;
```

Exp 8

a)

```
#include <iostream>
using namespace std;
class mstring {
        string str;

public:
    mstring (string s)
    { str = s; }

    mstring (){
        str = " ";
    }
void operator + (mstring obj)
    { str = str + obj.str; }

void disp(){
    cout << str;
  }
};
int main(){
    mstring s1("xyz"), s2("sqr"), s3;
    s1 + s2;
    cout << "Concatenated string: ";
    s3 = s1;
     s3.disp();
}
```

b)
```
#include <iostream>
using namespace std;
class ilogin {
    protected:
        string name;
        string pass;
public:
  virtual void accept(){
   cout<<"Enter Name and Password: ";
   cin >> name >> pass;
  }
virtual void disp(){
   cout<<"Name" <<name <<"Password" << pass;
   }
};
class elogin: public ilogin {
        string email;
        string pass;
 public:
    void accept(){
    cout<<"Enter email and Password: ";
    cin >> email >> pass;
    }
   void disp(){
   cout <<"Email"<< email <<"Password" <<pass;
    }
   };
class mlogin : public ilogin {
    string mid;
    string pass;
```

```
public:
 void accept(){
  cout << "Enter m_id & password: ";
 cin >> mid >> pass;
 }
 void disp(){
 cout << "M_id " << mid << " Password " << pass;
   }
};
int main(){
   ilogin *iptr;
   ilogin i;
   elogin e;
   mlogin m;

 iptr = &i;
   iptr -> accept();
   iptr -> disp();
   cout << endl;

  iptr = &e;
  iptr -> accept();
    iptr -> disp();
   cout << endl;

 iptr = &m;
 iptr -> accept();
 iptr -> disp();
 cout << endl;
```

5/11

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

int main(){
  ifstream fin("source.txt");
  ofstream fout("destination.txt");

  if(!fin || !fout){
   cout<<" Error opening files.... \n";
   return 1;
   }
  cout<<" File opened successfully \n";

// copy file content.
 char ch;
 while (fin.get(ch)){
       fout.put(ch);
    }
 fin.close();
 fout.close();

//count words
 fin.open("source.txt");
 int wordcount = 0;
 string word;
 while (fin >> word){
      wordcount++;
    }
 fin.close();
 cout<<"word count : "<< wordcount <<"\n";
```

```
//count specific word occurrence
fin.open("source.txt");
string target = "world";
int count = 0;
while (fin >> word) {
    if (word == target)
    { count++;
    }
fin.close();
cout<<"word occurrence: "<< count << "Ts" "\n

//count digit & spaces.
fin.open("source.txt");
int digitcount = 0, spacecount = 0;
while (fin.get(ch))
    {
      if (isdigit(ch))
        digitcount++;
      if (isspace(ch))
        spacecount++;
      }
  fin.close();

cout <<" Digits: "<< digitcount <<"\n";
cout <<"Spaces: "<< spacecount <<"\n";

return 0;
}
```

Experiment 10

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

template <class T> T sum(T a[], int n){
        T sum = 0;
    for (int i=0; i<n; i++){
            sum = sum + a[i];
        }
        return sum;
}
int main (){
    int n=5;

  int a[n];
  float b[n];
  double c[n];

cout<<"Enter 5 integer values: ";
 for (int i=0; i<n; i++){
        cin>> a[i];
    }
cout<<"Enter 5 float values: ";
 for (int i=0; i<n; i++){
        cin>> b[i];
    }
cout<<"Enter 5 double values: ";
   for (int i=0; i<n; i++){
    cin>> c[i];
    }
cout << sum<int> (a,n) <<endl;
cout << sum <float> (b,n) <<endl;
cout << sum <double> (c,n) << endl;
```

E:

b)
```
#include <iostream>
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 s;
cout<<" Enter an integen: ";
 cin>> num;
cout<<" Enter a string: ";
 cin>> s;

cout<< square <int> (num) <<endl;
cout << square <string> (s) <<endl;

return 0;
}
```

c) #include <iostream>
#include <cmath>
using namespace std;

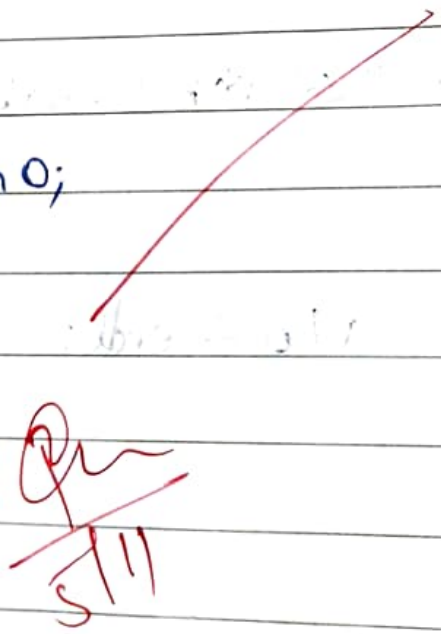
template <class T> class Calculator {
    public:
    T a, b;

void accept(){
    cout<<"Enter values of a & b: ";
    cin>>a>>b;
    }
void add(){
    cout<<"Addition = "<<a+b<<endl;
    }
void sub(){
    cout<<"Subtraction = "<<a-b<<endl;
    }
void mul(){
    cout<<"Multiplication = "<<a*b<<endl;
    }
void div(){
    cout<<"Division = "<<a/b<<endl;
    }
};
int main(){
  Calculator <double> cal;
   int c;

cal.accept();

```
while (1){
cout<<"1. Addition\n 2. Subtraction\n 3. Multiplication\n 4.
      5. Exit\n";
cout<<"Enter your choice: ";
cin>>c;

switch(c){
  case 1: cal.add();
  break;
  case 2: cal.sub();
  break;
  case 3: cal.mul();
  break;
  case 4: cal.div();
  break;
  case 5: exit(0);
  default: cout<<"Invalid input...\n";
  break;
  }
}
return 0;
}
```



Experiment 11

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

int main () {
  vector <int> vec(5);
   int i;
cout <<"Enter vector 5 elements: ";
for (i=0; i<5 ; i++) {
     cin >> vec[i];
  }
  cout <<endl;
  cout <<"Vector elements are: " <<endl;
for ( i = 0 ; i<5; i++) {
     cout << vec[i] << endl;
   }

cout<<"modified elements are: ";
   for (i=0; i<5 ; i++) {
     vec[i] = vec[i] + i*2;
    }
    for (i=0; i<5 ; i++) {
      cout << vec[i]<<" ";
     }
  cout <<endl;
int scalar;
cout <<"Enter a scalar value to multiply: ";
  cin >> scalar;
cout <<"After multiplying by scalar: ";
 for (i = 0 ; i<5 ; i++)
    vec[i] = vec[i] * scalar;
```

```
for(i=0; i<5; i++)
 cout << vec[i] << " ";
 cout << endl;
}
```

* Experiment 12

a)

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

void prints (stack <string> s) {
      if (is.empty()) {
        cout<<"Stack is empty\n";
       } else {
          cout <<"Stack elements are : \n\n";
          while (!s.empty()) {
             cout << s.top() << endl;
             s.pop();
          }
        cout << endl;
      }
   }
int main() {
   int ch;
   string name;
   stack <string> games;

  while (1)
  {
   cout<<"\n1.Push \n2.Pop \n3.Pop one \n4.Size
```

```
\n5. Display \n6.Exit\n" ;
cout <<" Enter choice: \n";
 cin >> ch;
 switch (ch)
 {
  case 1:
  cout <<" Enter Game Name: " <<endl;
  cin >> name;
  games.push (name);
  break;

 Case 2:
 cout << "Popped All Elements: \n\n " << games.top();
     while( !games. empty ()){
          cout << games.top()<< endl;
          games . pop();
        }
        cout << endl;
        break;

 Case 3:
 cout <<"Popped one elements: " <<games.top() <<endl;
  games.pop();
  break;

 Case 4:
 cout <<" Stack size: " << games . size ()<< endl;
 break;
```

```
case 5:
 print(games);
 break;

case 6:
    exit(0);
   break;

default:
 cout << "Invalid choice!\n";
    break;
   }
  }
}
```

b)
```
#include <iostream>
#include <queue>
using namespace std;

void prints (queue<string> q) {
    if (q.empty()) {
    cout << "Queue is empty\n";
  } else {
     cout << "Queue elements are: \n\n";
    while (!q.empty()) {
       cout << q.front() << endl;
       q.pop();
     }
    cout << endl;
    }
  }
int main (){
    int ch;
    string name;
    queue <string> games;

 while(1)
{
cout << "\n1. Push \n2. Pop \n3. Pop one element \n4. Size
     \n 5. Display \n6. Exist Exit \n";
cout<<" Enter choice: \n";
 cin >> ch;

switch(ch)
{
```

```
case 1:
 cout << "Enter game name: " << endl;
 cin >> name;
 games.push(name);
 break;

case 2:
 cout << "Popped All Elements: \n\n";
 while (!games.empty()) {
   cout << games.front() << endl;
   games.pop();
 }
 cout << endl;
 break;

case 3:
 cout << "Popped one element: " << games.front() << endl;
 games.pop();
 break;

case 4:
 cout << "Queue size: " << games.size() << endl;
 break;

case 5: prints(games);
        break;
case 6: exit(0);
        break;
default: cout << "Invalid choice!\n";
        break;
    }
  }
}
```

Q no 5 ||