classmate
Date
Page

Experiment 1:

a)

```
#include <iostream>
using namespace std;
class student
{   string name;
    int rollno, cls;
  public:
  void input()
  {  cout << "Enter name of student";
     cin >> name;
     cout << "Enter rollno. of student";
     cin >> rollno;
     cout << "Enter the class";
     cin >> cls;
  }
  void display()
  {
    cout << "Name : " << name << "In Roll no.
            : " << rollno << "In Class : " <<
            cls << endl;
  }
} s1;
  void main ()
  {
    s1. input();
    s1. display();
  }
```

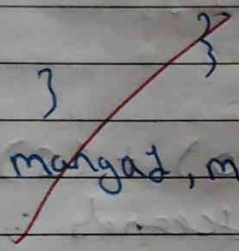
Output

```
Enter name of Student : Smarty
Enter Roll number : 22
Enter class : 10

Name : Smarty
Roll no. : 22
Class : 10
```

b)

```
#include <iostream>
using namespace std;
class Book {
    string name;
    int pages;
public:
float price;
void input()
{ cout << " Enter Name of Book:";
    cin >> name;
    cout << " Enter Number of pages of book:";
    cin >> pages;
    cout << "Enter the price of Book :";
     cin >> price;
}
int getprice()
 { return price; }
```

```
void Book withgreaterprice (Book & b1, Book &b2)
{
    if ( b1.getprice( ) > b2.getprice( ))
      {
        cout <<"Book with greater price is :"<<
              b1.name;
      }
      else {
            cout <<" Book with greater price
                  is :"<< b2.name;
```



```
}
} manga1, manga2;
int main( )
{ Book manga;
  manga1.input ( );
  manga2.input ( );
  manga.bookwithgreaterprice (manga1, manga2);
}
```

Output

```
Enter Name of Book : Ikigai
Enter Number of pages of Book : 348
Enter the price of Book : 985
Enter name of Book : Deadly Sins
Enter Number of pages of Book : 225
Enter the price of Book : 199

Book with greater price is : Ikigai
```

c)
```
#include <iostream>
using namespace std;
class time
{ int H, M, S;  char d1, d2;
  public:
    void input()
    { cout << "Enter time in HH:MM:SS format";
      cin >> H >> d1 >> M >> d2 >> S;
    }
    void calculate()
    {
      int total;
      total = (H * 3600) + (M * 60) + (S);
      cout << "Total time in seconds : " << total;
    }
} t;

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

Output:

Enter time in HH:MM:SS format : 1:30:05

Total time in seconds : 5405.

30/7/25

Date
Page

Experiment 2

a)

```
#include <iostream>
using namespace std;
class city
 {
  int pop[5]; string name [5];
  public:
   void input ()
   {
     for(int i=0; i<5;i++)
      {
        cout <<" Enter name & poppulation of
                  city : ";
        cin >>name[i]>>pop[i];
      }
   }
   void display ()
   { int max=pop[0]; int k;
     for(int i=0; i<5; i++)
       {
         if (max<pop[i])
           {
             max=pop[i];
             k=i;
           }
       }
cout<<" City with greater population is :"<<
      name[k] <<" pop[k];
  } }c1;
int main() {
  c1.input();
  c1. display(); }
```

b)
```
#include <iostream>
using namespace std;
class acc {
    int acc_no[3];
    float balance[3];
    public:
    void input ( )
    { for(int i=0; i<3; i++) {
      cout << "Enter account number & Balance :";
      cin >> acc_no[i] >> balance[i];
    }
    }
    void calculate ( )
    {
        for(int i=0; i<3; i++)
        {
          if ( balance[i] >=5000 )
          {
            balance[i] = balance[i] + (balance[i]*10*1/
          cout << " Balance after Intrest : " <<
          cout << acc_no[i] << balance[i];
          }
        }
    }
} ac;
int main ( )
{
  ac.input ( );
  ac.calculate ( );
}
```

c)
```
#include <iostream>
using namespace std;
class staff
{
    string name[5];
    sting post[5];
    public:
    void input()
    {   for(int i=0; i<5; i++)
        {
          cout<<" Enter Name and Post : ";
          cin>> name[i]>> post[i];
        }
    }
    void display()
    {
       for(int i=0; i<5; i++)
        {
          if (post[i] == "HOD")
           {
             cout<<" Name : " << name[i] << " Post:"
                 << post[i] ;<<endl;
           }
        }
    }
} sf;
int main()
{
   sf. input();
   sf. display();
}
```

30/7/25

Experiment 3 :

Q)
```
#include <iostream>
using namespace std;
class Book{
string auth_name;
string book_title;
float price;
public:
void input()
{ cout<<"Enter Name of Author, Book
        Title and Price of Book";
    cin >> auth_name >> book_title >> price;
}
void display()
 {
    cout<<"Book - Title : "<< book_title <<endl
        <<"Author Name : "<<auth_name<<
        endl <<"Price : "<<price;
 }
};
int main()
{
    Book *p;
    p=&B;
    p -> input();
    p-> display();
}
```

b)

```
#include <iostream>
using namespace std;
class student {
 int roll;
 float percentage;
public:
 void accept()
  {  cout << Enter Roll Number : " << endl;
     cin >> this -> roll;
     cout " Percentage : " << endl;
     cin >> this -> percentage;
  }
 void display()
  {
    cout << " Roll No : " << this -> roll <<
     endl << " Percentage : " << this -> percentage
        << endl;
  }
 } s;
 int main()
 {
   s.accept();
   s.display();
 }
```

c) #include <iostream>
using namespace std;
class outer { public:
void display ( )
{
cout << " From Outer class ";
}

class inner{

void displayy ( )
{
cout << " From Inner class ";
}

};
};

int main ( )
{
outer :: inner in ;

out. display( );
in. displayy( );
}

13/8

Experiment 4

1]
```
#include <iostream>
using namespace std;
class Num {
    int a;
public:
    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 << " Num 1 = " << n1.display << endl
        << " Num 2 = " << n2.display;
    n1.swap (n2);
    cout << " num 1 = " << n1.display ();
    cout << " num 2 = " << n2.display();
    }
```

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

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

3)
```
#include <iostream>
using namespace std;
class B;
class A {
    int a;
    public:
    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);
 }
```

4)
```
#include <iostream>
using namespace std;
class result2;
class result1 {
  public:
    int o;
    void accept()
    {
      cout << "Enter value of o :";
      cin >> o;
    }
  friend void avg (result1 r1, result2 r2);
} r1;
class result2 {
public:
int i;
void accept()
  { cout << "Enter value of i :";
    cin >> i; }
}
friend void (result1 r1, result2 r2);
} r2;
void avg (result1 r1, result2 r2) {
  float avg;
  avg = (r1.o + r2.i)/2;
  cout << "Average = " << avg;
}
int main () {
r1.accept (); r2.accept();
avg (r1, r2);
}
```

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);
 }
```

Experiment 5

1].

```
#include <iostream>
using namespace std;
class SumCal
{
  int n,sum=0;
  public:
// void accept()   not needed.
  SumCal(int num)
  {
    n=num;
    for(int i=0; i<=n; i++)
    {
      sum+=i;
    }
  }
  void display()
  {
    cout<<"Sum = "<<sum;
  }
};
int main()
{ SumCal c1;
  SumCal s(5);
  s.display();
  SumCal c2(s);
  c2.display();
}
```

```
SumCal(){
  n=6;
for(int i=0; i<=n; i++)
 { sum+=i; }
}
SumCal(SumCal &s1)
{  n=s2.n;
for(int i=0; i<=n; i++)
{ sum+=i;
}
}
```

2)
```
#include <iostream>
using namespace std;
class Student
{
    char name[50];
    float per;
    public:
    Student()
    {
     cout<<" Name & Percentage : ";
     cin >> name >> per;
    }
    void display()
    {
     cout<<" Name : " << name <<endl;
     cout<<" Percentage : " <<per <<endl;
    }
    Student (string n, float p)
    {
     name = n;
     per = p;

    cout <<"Name : " <<name <<endl <<" Percentage :
        <<per <<endl;
    }
    Student( student &s ){
    name = s.name;
    per = s.per;
    cout <<" Name and Percentage : " <<name <<per <<end
    }
```

```
int main()
{
  Student K;
  K.display();
  Student S("Chirag", 90.8);
  Student Z(S);
}
```

3]
```
#include <iostream>
using namespace std;
class College {
 int rollno; string name, course;
 public:
College(int r, string n){
 rollno = r;
 name = n;
 course = "Computer Engineering";
}
void disp(){
cout<<"Name : "name<<endl<<"Roll No : "<<rollno
 <<endl<<"Course : "<<cource;
 }};
int main()
{ Student s("Shreyas", 21);
   Student S1("Chirag", 9);
    S.disp();
    S1.disp();
}
```

4]

```
#include <iostream>
using namespace std;
class Cons {
int x, y;
public:
Cons () {
 x = 0;
 y = 0; }
Cons (int a) {
   x = a;  y = 0;
 }
Cons (int a, int b) {
    x = a;  y = b;
 }
 void display ( ) {
 cout << " x = " << x << " y = " << y << endl;
}};
int main( )
{
  Cons d1;
  Cons d2 (10);
  Cons d3 (20, 30);

d1.display ( );
d2.display( );
d3.display( );
}
```

24/9/25

classmate
Date
Page

Experiment 6

Q1] Single inheritance

→
```
#include <iostream>
using namespace std;
class person{
        protected:
        string name;
        int age;
      }
class info : protected person{
   string gender;
    public:
   void accept (){
   cout<<"Enter Name, Age and gender : ";
   cin >> name >>endl >> age >>endl >> gender >> endl;
   }
void display ( ) {
 cout <<" Name : " << name << " In Age : " <<age<<
   " In Gender :" << gender <<endl;
   }
 } i;
int main ( ){
  i. accept ( );
  i. display ( );
}
```

Q2] Multiple Inheritance

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

Q3) Multi-level Inheritance

→
```
#include <iostream>
using namespace std;
class vehicle {
        public:
        string brand, model;
      }
class car : protected vehicle {
        protected:
        string attribute;
    }
class ecar : protected car {
        int battery;
        public:
        void accept() {
        cout<<" Enter Brand and model of car:";
        cin>> brand >> model >> endl;
        cout<<" Enter type of car : ";
        cin>> attribute >> endl;
        if(attribute == "ecar")
         {
            cout<<" Enter Battery capacity in MAH";
            cin>> battery;
         }
      }
      void display() {
  cout<<" Brand : "<< brand <<endl<<" Model : "<<
    model <<endl<<" TYPE : "<<attribute <<endl;
    if( attribute == " ecar" )
     {  cout<<"Battery Capacity in (MAH) ":<<
            battery ; } } e;
```

```
int main() {
 e.accept();
 e.display();
}
```

Q4) Hierarchial Inheritance

→
```
# include <iostream>
using namespace std;
class Dept {
    protected:
    string dept;
  }
class students {
 class teacher : protected Dept {
     string tname;
     int id;
     public:
    void accept(){
    cout<<"Enter Department :";
    cin>>dept>>endl;
    cout<<"Enter name of teacher and id :";
    cin>>tname>>id>>endl;
    }
    void display(){
    cout<<"Department :<<dept<<endl<<"Teacher
     Name :"<<tname<<endl<<"ID :"<<id<<endl;
   }
 } t;

class student : protected dept {
        string sname;
        int roll;
```

```
public:
void accept(){
cout<<"Enter name of student and roll no: ";
cin>>sname>>roll;
}
void display(){
cout<<"Name of student : "<<sname<<endl;
 cout<<" Roll NO. : "<<roll;
}
} s;
int main(){
t.accept();
s.accept();
t.display();
s.display();
}
```

Q.5] Hybrid Inheritance

```
#include <iostream>
using namespace std;
class student {
    protected:
    string name;
    int roll;
    }
class academic : protected student {
        protected:
        int marks;
        }
class sports : protected student, protected academic
    {       protected:
            int score;
    }
class result : protected sports {
        int total;
        public:
        void accept() {
    cout<<" Enter name & roll no. of student:";
     cin>> name>> roll >>endl;
      cout<< " Enter academic and sports score";
       cin>> marks >> score >>endl;
    }
  void cal(){
   total = score + marks;
   cout <<"Name : "<<name << " In Roll No. : "<<roll<<
    " In Total Score : " << total <<endl;
  }
} r;
```

```
int main ( ) {
r.accept( );
r.cal( );
}
```

6] Virtual Base class :

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

```
int main ()
{
  result r;
  r.print ();
}
```

26/9/25

Experiment 7.

a]

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

b]
```
#include <iostream>
using namespace std;
class S {
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 numbers : "<<s<<endl
}
void sum (int b[10]){
 int s=0;
for(i=0; i<10; i++) {
 s+= b[i];
}
cout<<" Sum of 10 integers : "<<s<<endl;
}
} S1;
int main () {
 float c[5]; int d[10];
cout<<"Enter five float nos : \n";
for(int i=0; i<5; i++){
 cin>> c[i] ; }
cout<<" Enter 10 integer nos : \n";
for(int i=0; i<10; i++) {
 cin>> d[i]; }
 S1.sum(c);
 S1.sum(d);
}
```

c]
```
#include <iostream>
using namespace std;
class num{
int a;
public:
void accept(){
cout<<" Enter value of a: ";
cin>>a;
}
void disp(){
cout<<" value of a :"<<a;
}
 void operator -(){
    a=-a;
}
} n1;
int main(){
n1.accept();
-n1;
n1.disp();
}
```

d]
```
#include <iostream>
using namespace std;
class num {
 int a;
public:
 void accept () {
 cout << " Enter value of a : ";
 cin >> a;
 }
 void disp () {
 cout << " value of a : " << a ;
 }
 void operator ++ () {
  a = ++ a;
 }
} n1;
int main () {
 n1. accept ();
 ++n1;
 n1. disp ();
}
```

30/10/25

Experiment 8

8]
```
#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("pqr"), s3;
 s1+s2;
 cout << " concatenated string : ";
 s3 = s1 ;
 s3.disp ( );
}
```

2]
```
#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 <<endl;
}
};
class elogin : public ilogin {
    string email;
    string pass;
  public:
  void accept() {
  cout<<" Enter Email & Password : ";
  cin>> email >> pass >> endl;
  }
void disp () {
cout<<" Email : "<<email <<" Password : "<< pass <<endl;
}
};
class ulogin : public ilogin {
 string uid;
 string pass;
```

```
public :
void accept ( ) {
cout << " Enter U_id and password : ";
 cin >> uid >> pass;
}
void disp ( ){
cout << " U_id : " << uid << " Password " << pass ; }
} ;
int main ( ){
ilogin *iptr ;
 ilogin i;
elogin  e;
 ulogin  u;
iptr = &i;
iptr -> accept ( );
iptr -> disp( );
cout<<endl;
 iptr=&e;
iptr -> accept ( );
iptr -> disp( );
cout<<endl;
 iptr = &u;
iptr -> accept ( );
iptr -> disp( );
 cout<<endl;
}
```

30/10/25

Experiment 9

* Write a program to perform various operation on file.

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

int main() {
ifstream fin;
ofstream fout;

fin.Open("source.txt");
fout.open("destination.txt");
if (!fin) {
  cout<<" Error opening source file \n";
  return 1;
}
char ch;
while (fin.get(ch)) {
    fout.put(ch);
  }
fin.close();
fout.close();
cout<<" File copied successfully \n";
fin.open("source.txt");
int wordcount = 0;
sting word;
while (fin>> word) {
 wordcount++;
}
```

```
cout << " Word Count : " << wordcount << "\n";
fin.close();
fin.open("source.txt");
int count = 0; string target = "World";
while ( fin >> word ) {
  if ( word == target )
    {
      count++;
    }}
cout << " Word Occurance : " << count << "\n";
fin.close();
int digitcount=0;
int spacecount=0;
fin.open("source.txt");
while( fin.get(ch) ){
   if( isdigit(ch) ){
       digitcount++;
   }
   if(isspace(ch)){
      spacecount++;
   }
}
cout<<" Digits : "<<digitcount<<"\n";
cout<<" Blanks : "<<spacecount<<"\n";
}
```

5/11

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+= a[i];
  }
  return sum;
}
int main ( ){
 int n=5;
int inta [n];
float floata [n];
double doublea [n];

Cout<<"Enter 5 integers : \n";
for(int i=0; i<n; i++)
  {
    cin>> inta[i];
  }
cout <<"\nEnter 5 floats :";
for(int i=0; i<n; i++ ) {
   cin>> floata [i];
  }
cout<<"\nEnter 5 doubles :";
 for (int i=0; i<n; i++)
  {
    cin >> doublea [i];
  }
```

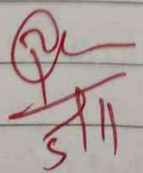
```
cout << "\n Sum of Integers : " << sum(inta, n);
cout << "\n Sum of Floats : " << sum(floata, n);
cout << "\n Sum of Doubles : " << sum(doublea, n);

}
```

b]

```
#include <iostream>
#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 a string and a number : ";
cin>> str >> num>>endl;

cout<< "\n Square of Integer:" << square(num);
cout << "\n Square of String : " << square(str);

}
```

c]
```
#include <iostream>
using namespace std;
template <class T>
class calculator {
 T a, b;
 public:
 calculator(T n1, T n2){
   a=n1;
   b=n2;
  }
 void add( ) {
 cout<<" Addition : "<<a+b<<endl;
 }
 void sub( ) {
 cout <<" substraction : " << a-b <<endl;
 }
 void mul( ) {
 cout<<" Multiplication : "<<a*b<<endl;
 }
 void divide ( ) {
 if ( b!=0)
 { cout<<" Division : "<<a/b <<endl; }
 else {
    cout<<" Division by zero not possible "<<endl;
  }
 }
 };
```

```
int main ( ){
double x, y, z ; double x, y;
cout <<" Enter two numbers :";
 cin >> x >> y >>endl;
int z;
 calculator <double> cal (x, y);

while (1) {
cout<<" 1. Addition \n 2. Substraction \n 3. Multipl
        \n 4. Divide \n 5. exit <<endl;
cin >> z;

switch (z) {
case 1 :  cal. add( );   break;
case 2 :  cal. sub( );   break;
case 3 :  cal. mul( );   break;
case 4 :  cal. divide( );  break;
case 5 :  exit (0);
 }
}
return 0;
}
```



CLASSMATE
Date
Page

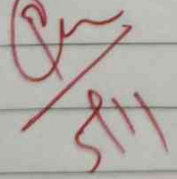
Experiment 11

```
#include <iostream>
#include <vector>
#include <ctype>
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;
```

```
for (i=0; i<5; i++)
{
    vec[i] = vec[i] * scalar;
}
cout<<"After Multiplying by scalar : ";
for(i=0; i<5; i++) {
 cout<< vec[i] <<"   ";
} cout <<endl;

}
```

With iterator

```
vector <int> :: iterator it;
cout<<"Vector elements using iterator :"
for( it = vec.begin() ; it != vec.end() ; ++i
{
   cout<< *it <<endl;
}
return 0;
}
```



Experiment 12

a] Implement Stack

```
#include <iostream>
#include <stack>
using namespace std;
void print (stack <string> s) {
 if (s.empty()){
  cout<< "Stack is Empty \n"; }
else {
   cout << " Stack elements are : \n";
   while( !s.empty() ) {
    cout << s.top() << endl;
    s.pop();
  }
 cout<<endl;
 }3
 int main() {
 int ch; string name;
Stack <string> games;
while (1) {
 cout<< "\n1. Push \n 2. POP All elements \n3. POP
 on element \n 4. Size \n 5. Display Stack
 \n 6. 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 Element : "games.top()<<en
games.pop();
break;
Case 4:
cout<<" STACK Size : "games.size()<<endl;
break;

Case 5:
prints(games);
break;

Case 6:
Exit(0);
break;

default:
cout<<" Invalid input! ";
  break;
}
}
}
```

b] Implement Queue

```
#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 ";
 while ( ! q.empty( )){
 cout << q.front() <<endl;
  q.pop();
 }
cout<<endl;
}
int main () {
int ch;
string name;
queue <string> games;
while(1) {
cout <<" \n 1. Push \n 2. POP All elements \n 3. Pop
1 element \n 4. Size \n 5. Display Queue \n 6.
 Exit ";
Cout <<" Enter your choice : ";
 cin >> ch ;

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

```
case 2:
cout<<"Popped All elements : \n";
while (! games.empty()){
 cout<<games.front()<<endl;
 games.pop();
}
cout<<endl;
break;
Case 3:
Cout<<"Popped One Element : "<<games.front()
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;
 }
}
}
```