Sarthak Kale

@kalesarthak666

Follow

Badges

Certifications

kalesarthak666 has not earned any certificates yet.

Work Experience

kalesarthak666 has not updated employment details yet.

Education

kalesarthak666 has not updated education details yet.

Links

kalesarthak666 has not updated links details yet.

Skills

kalesarthak666 has not updated skills details yet.

PAGE NO.
DATE / /

experiment — 1

1) write a c++ code to declare a class student having data members as roll no and name accept and display data for single student

ans)
```
#include<iostream.h>
using namespace std;
class student
{   int roll-no,
    string name;
    public:
    void accept()
    { cout<<"enter the student name & roll no:";
      cin>>"name">>"roll-no"; }
    void disp()
    { cout <<"student name:"<<name;
      cout <<"\n student roll-no:"<<roll-no; } };
int main()
{ student s1;
  s1.accept();
  s1.disp();
  return 0; }
```

2) write a c++ code to create a class book having data members as book_name, book_price, & book_id. Accept the data for 2 books & display the name of the book having greater prices.

ans)
```
#include <iostream>
using namespace std;
class class_book
{ string book_name,
  int book_id;
  public:
  int book_price;
```

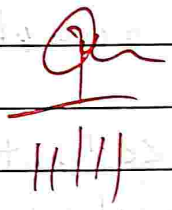
```
void accept()
{
  cout<<"enter the book name,price & id:";
  cin>>book_name>>book_price>>book_id;}
void disp()
{ cout<<"\n book name:"<<book_name;
  cout<<"\n book price:"<<book_price;
  cout<<"\n book ID:"<<book_id;}};
int main()
{ class_book B1,B2;
  B1.accept();
  B2.accept();
  if(B1.book_price > BB2.book_price)
  { B1.disp()
  else
  { B2.disp();}
  return 0;}
```

3) write a C++ program to declare a class time having data members as H, M and S. Accept data for one object and display total time in seconds

ans)

```
#include<iostream>
using namespace std;
class time
{ int H,M,S;
  public:
  void accept()
  {cout<<"enter the time in hours,minutes & seconds:";
    cin>>H>>M>>S;}
  void calculate()
  { H = H*3600;
    M = M*60;
```

```
S = S + H + M; }
void disp()
{ cout << " total time in seconds = " << S; }}
int main()
{ Time T1;
  T1.accept(); T1.calculate();
  T1.disp();
  return 0; }
```



PAGE NO.
DATE / /

experiment -2

1) WAP to declare a class 'city' having data members as name & population. Accept this data for 5 cities & display name of city having highest population.

ans)
```
#include<iostream>
using namespace std;
class city
{ public:
  string name;
  int population;
  void accept()
  {
  cout<<"enter city name:";
  cin>>city;
  cout<<"enter city population:";
  cin>>population; }
  void disp()
  { cout<<"city having highest population:"<<name;}};
  int main()
  { city c[5];
   int i, max;
   for(i=0; i<5; i++)
   { c[i].accept(); }
   max = c[0].population;
   for = (i=0; i<5; i++)
   { if( c[i] population>max)
     { max=i }}
     c[max].disp();
   return 0; }
```

PAGE No.
DATE / /

2) WAP to declare a class 'account' having data members as account & balance. Accept this is data for 10 accounts & give interest of 10% where balance is equal or greater than 5000 & display them.

ans)
```
#include <iostream>
class account
{ public:
  int acc_no;
  float balance;
  void accept()
  { cout<<"enter account number:";
    cin>>"acc_no";
    cout<<"enter account balance:";
    cin>>"balance";}
  void disp()
  { cout<<"\n account number:"<<acc_no;
    cout<<"\n account balance:"<<balance;}};
int main()
{ account A[10];
  int i;
  for (i=0;i<10;i++)
  { A[i].accept();}
  for (i=0;i<10;i++)
  { if (A[i].balance>=50000)
  { A[i].balance =A[i] balance + (0.1*A[i]);
    A[i].balance();
    A[i].disp();}}
  return 0;}
```

3) WAP to declare a class 'staff' having data members as name & post. Accept this data for 5 staff and display name of staff who have "HOD"

ans)
```
#include <iostream>
using namespace std;
class staff
{ string name;
  public:
  string post;
  void accept()
  { cout<<"enter the staff name : ";
    getline(cin,name);
    cout<<"enter the staff post:";
    getline(cin,post);}
    void disp()
  { cout<<"\n staff name:"<<name;
    cout<<"\n staff post:"<<post;}};
int main()
{staff s[5];
  int i;
  for(i=0; i<5; i++)
  { s[i].accept();}
  for(i=0; i<5; i++)
  { if (s[i].post =="HOD")
  { s[i].disp();}}
  return 0;}
```

PAGE No.
DATE / /

experiment 3

1) WAP to declare a class 'book' containing data members as book_title, author name & price. Accept & display the information for one object using a pointer to that object.

ans)
```
#include<iostream>
using namespace std;
class book
{ string book title;
  string author name;
  int price;
  public:
  void accept()
  { cout<<"enter book title:";
    getline(cin, booktitle);
    cout<<"enter author name:";
    getline(cin, authorname);
    cout<<"enter book price:";
    cin>>price;}
  void disp()
  { cout<<"Book title:"<<book title;
    cout<<"\nauthor name:"<<authorname;
    cout<<"\n book price:"<<price;}};
  int main()
  { book B1;
    book *p;
    p=&B1
    p->accept();
    p->disp();
    return 0;
```

2) WAP to declare a class 'student' having

ans)
```
#include<iostream>
using namespace std;
class student
{ int roll_no;
  float percentage;
  public:
  void accept()
  { cout<<"enter the student roll no:";
    cin>>roll_no;
    cout<<"enter student percentage:";
    cin>>percentage;}
  void disp()
  { this->accept();
    cout<<"\n roll no. of the student:"<<roll_no;
    cout<<"\n percentage of the student:"<<percentage;
  }};
int main()
{ student s1;
  s1.disp()
  return 0;
```

3) WAP to demonstrate the use of nested class?

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

```
void input student details()
{ cout<<"enter student's name:";
  getline(cin,nam);
  cout<<"enter student's roll number:";
  cin>>roll-no;
}
void display student details()
{ cout<<"\n student name:"<<name<<endl;
  cout<<"\n student roll-no:"<<roll-no<<endl;
}
class marks
{ private:
  int science;
  int english;
  public:
  void input marks()
  { cout<<"enter marks in science:";
    cin>>science;
    cout<<"enter marks in english:";
    cin>>english;
  }
  void display marks()
  { cout<<"Marks in science:"<<science<<endl;
    cout<<"marks in english:"<<english<<endl;
  }
};
int main()
{ student s1;
  student :: marks M1;
  s1.input student details();
  M1.input marks();
```

```
S1.display student details();
M1.display marks();
return 0;
}
```

PAGE No. / DATE / /

expt - 4

Q) write a c++ program to solve the following problem statement & show output:

1) swap 2 numbers from same class using object as functions & write swap functions as member functions.

ans) code:

```
#include <iostream>
using namespace std;
class num{
class public:
    int n1,n2;
    void accept(){
        cout<<"enter first number:"<<endl;
        cin>>n1;
        cout<<"enter second number:"<<endl;
        cin>>n2;
    }
    void display(){
        cout<<"Num1."<<n1<<endl;
        cout<<"Num 2."<<n2<<endl;
    }
    void swap(num&){
        int temp=n1; n1=n2; n2=temp;
        cout<<"swapped:\n first number:"<<n1<<"\n
            second number:"<<n2;
    }
};num nu;
int main(){
    nu.accept();
    nu.display();
    nu.swap(nu);
    return 0;
}
```

O/P

```
enter first number:
5
enter second number:
9
num1 : 5
num 2 : 9
swapped
first number : 9
second number : 5
```

2) Swap 2 number from same class using concept of friend function.

ans)
```
#include<iostream>
using namespace std;
class num{
public: int n1, n2;
    void accept(){
        cout<<"enter first number:"<<endl;
        cin>>n1;
        cout<<"enter second number:"<<endl;
        cin>>n2;
    }
    void display(){
        cout<<"num1:"<<n1<<endl<<"num2:"<<n2<<
    }
}nu;
void swap(num){
    int temp = nu.n1;
    nu.n1 = nu.n2;
    nu.n2 = temp;
    cout<<"swapped:\n first:"<<nu.n1<<"second:"<<nu.
        <<endl;
}
int main(){
    nu.accept();
    nu.display();
    swap(nu);
    return 0;
}
```

O/P

```
enter first number
5
enter second number
9
num 1: 5
num 2: 9
swapped
first: 9
second: 5
```

PAGE NO.
DATE / /

3) swap 2 number from different class using friend function.

ans)
```
#include<iostream>
using namespace std;
class num1 {
public: int n1;
   void accept(){
        cout<<" enter first number: "<<endl;
        cin >> n1;}
   void display(){ cout << "num1: " <<n1<<endl;}
}nu1;
class num2 {
public: int n2;
   void accept(){
      cout <<" enter second number: "<<endl;
      cin>>n2;}
   void display(){cout<<"num2: "<<n2 <<endl;
}nu2;
void swap (num1, num2){
        int temp = nu1.n1;
        nu1.n1 = nu2.n2;
        nu2.n2 = temp;
        cout<<"swapped: \n first: "<<nu1.n1<<\n second:"
            <<nu2.n2<<endl;
}
int main(){
    nu1.accept();
    nu2.accept();
    nu1.display();
    nu2.&display();
    swap(nu1, nu2);
    return 0;
}
```

```
O/P: enter first number
     5
     enter second number
     9
     num1 = 5
     num2 = 9
     swapped:
     first: 9
     second: 5
```

4) create 2 classes result1 & result2 which stores the marks of students read the value of final for the class objects & ~~avg~~ compute avg.

ans)
```
#include<iostream>
using namespace std;
class result1{
public:
    int r1;
    void accept(){
        cout<<"enter first result:"<<endl;
        cin>>r1;}
}res1;
class result2{
public:
    int r2;
    void accept(){
        cout<<"enter second result:"<<endl;
        cin>>r2;}
}res2;
float avg(result1, result2){
    return (res1.r1+res2.r2)/2;
}
int main(){
    res1.accept();
    res2.accept();
    float avg = avg(res1, res2);
    cout<<"average:"<<avg;
    return 0;
}
```

o/p

```
enter first result:
95
```

enter second result:
99
average : 97

5) Find the greatest number among 2 numbers from 2 different classes using friend function.

ans)
```
#includ<iostream>
using namespace std;
class num1{
public: int n1;
    void accept(){
        cout<<"enter first number:"<<endl;
        cin>>n1;}
};nu1;
class num2{
public: int n2;
    void accept(){
        cout<<"enter second number:"<<endl;
        cin>>n2;}
};nu2;
int gnum(int n1, int n2){return (n1>n2?n1:n2);}
int main(){
    nu1.accept();
    nu2.accept();
    int gr=gnum(nu1.n1, nu2.n2);
    cout<<"greatest:"<<g<<endl;
    return 0;
}
```

O/P

enter first number:
5
enter second number:
9
greatest: 9

PAGE No.
DATE / /

Expt - 5

Q.1) WAP to find the sum of numbers between 1 to n using a constructor where the value of n will be passed to the constructo

ans) code:

```
#include <stdio.iostream>
using namespace std;
class sum{
    int n, sum;
    int total;
    public:
    sum(){          // 1. Default constructor
      n=0;
    total=0;
    cout<<"default constructor called"<<endl;
    }
    sum(int num){
        n=num;
        total=(n*(n+1))/2;
        cout<<"parameterized constructor called"<<
            endl;
    }
sum(const sum &S){
        n=S.n;
        total = S.total;
        cout<<"copy constructor called"<<endl;
    }
void display(){
    public:
        sum(int num)
        {
            n = num;
            sum = 0
```

```
        for (int i=0; i<n; i++)
        {
            sum+=i
        }
        cout << "sum=" << sum;
    }
};
int main ()
{
  sum s(10);
  return 0;
}
```

2) WAP to declare a class 'student' having data members as name & percentage write a constructor to initialize the data members. Accept & display data for one student

ans)

```
#include <iostream>
#include <string>
using namespace std;
class student
{
    private:
    string name;
    float percentage;
    public:
        student (string n, float p)
        {
            name = n;
            percentage = p;
            cout << "Name:";
            getline (cin, name);
            cout << "percentage:";
            cin >> percentage;
        }
```

```
void display()
{
    cout << "student name: " << name << endl;
    cout << "student percentage: " << percentage << endl;
}
};
int main(){
    student si("sarthak", 92.0)
    si.display()
    return 0;
}
```

3) Define a class 'college' members variables as roll_no, name, course, WAP using constructor with default value as "computer engineering" for course Accept this data for two object of class & display the data.

ans)
```
#include <string>
#include <iostream>
using namespace std;
class college
{
    private:
    string course_name, stud_name;
    int roll_no;
    public:
    college()
    {
        roll_no = 73;
        stud_name = "sarthak";
        course_name = "CSE";
    }
```

PAGE No.
DATE / /

```
void display()
{
    cout<<"Roll.no:"<<roll_no<<endl;
    cout<<"student name:"<<stud.name<<endl;
    cout<<"course name:"<<course_name<<endl;
}
};
int main()
{
  college c1;
  c1.display();
  return 0;
```

4) WAP to demonstrate constructor overloading

ans)
```
#include<iostream>
using name space std;
class rectangle
{
   int l,b;
   public:
   rectangle()
   {
       l=2;
       b=5;
   }
   rectangle(int x)
   {
       l=x;
       b=x;
   }
   rectangle(int x, int y)
   {
```

```
        l = x;
        b = y;
    }
    rectangle (rectangle &r3)
    {
        l = r3.l;
        b = r3.b;
    }
    void calculate ()
    {
        cout << " the area is : " << (l * b) << endl
    }
};
int main()
{
    rectangle r1
    r1.calculate();
    rectangle r2 (3);
    r2.calculate();
    rectangle r3 (6,6);
    r3.calculate();
    rectangle r4 (r3);
    r4.calculate();
    return 0;
}
```

Q4

4/4

expt - 6

DATE / /

1)a WAP to implement multilevel inheritance assume suitable data.

ans)
```
#include <iostream>
#include < string>
using namespace std;
class person
{
    protected;
    int age;
    string name;
};
class student: public person
{
    private:
    int roll_no;
    public:
    void accept()
    {
        cout <<" enter your name: ";
        cin >> name;
        cout <<" enter your age: ";
        cin >> age;
        cout <<"enter your roll no:";
        cin >> roll_no;
    }
    void display ()
    {
        cout <<" name: \n" << name;
        cout <<" Age: \n" << age;
        cout <<" Roll no: \n" << Roll_no;
    }
};
```

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

2) write a program to implement multiple classes assume suitable data

ans)

```
#include <iostream>
using namespace std;
class academic
{
    protected:
    int marks;
};
class sports
{
    protected:
    int score;
};
class result : protected academic, protected sports.
{
    int total_score = 0;
    public:
    void accept()
    {
        cout<<"enter the marks of the students:";
        cin >> marks;
        cout<<"enter the sports score of the students:";
        cin>> score;
    }
```

```
void calculate()
{
    total score = marks + score;
    cout<<"the marks of student: "<<marks<<endl;
    cout<<"The sports score of the student: "<<score<<endl;
    cout<<"the total score of the student: "<<total_score<<endl;
    }
};
int main()
{
    result r;
    r.accept();
    r.calculate();
    return 0;
}
```

2) WAP to implement hierarchial inheritance assume suitable ~~table~~ data.

ans)

```
#include<iostream>
using namespace std;
class vehicle
{
    protected
    string brand;
    int model;
};
class car: protected vehicle
{
    protected:
        string type;
};
class electric car: protected car
{
```

```
int battery capacity;
public:
void accept()
{
    cout<<"enter the brand of car:";
    cin>>brand;
    cout<<"enter the model of car:";
    cin>>model;
    cout<<"enter the type of car:";
    cin>>tyre;
    cout<<"enter the battery capacity of the
          car:";
    cin>>battery capacity;
}
void display()
{
    cout<<"\n The brand of the car:"<<brand<<endl;
    cout<<"\n The model of the car:"<<model<<endl;
    cout<<"\n The tyre of the car:"<<tyre<<endl;
    cout<<"\n The battery capacity of the car:"<<
        battery capacity<<endl;
    }
};
int main()
{
    electric car e;
    e.accept();
    e.display()
    return 0;
}
```

4) WAP to implement hybrid inheritance

```
#include <string>
#include <iostream>
using namespace std;
class person
{
    public:
    string name;
    int age;
    void get personal details()
    {
        cout<<"enter name:";
        cin>>name;
        cout<<"enter age:";
        cin>>age;
    }
    void show details()
    {
        cout<<"name:"<<name<<"age:"<<age<<endl;
    }
};
class student: public person
{
    public
    string course
    void get details()
    {
        cout<<"enter course:";
        cin>>course;
    }
    void show details()
    {
        cout<<"course:"<<course<<endl;
    }
```

```
};
class employee : public person
{
    public:
    string company;
    void get employee details ()
    {
        cout<<" enter company : ";
        cin>>company;
    }
    void show employee details ()
    {
        cout<<" company : "<<company <<endl;
    }
};
class intern : public student, public employee
{
    public:
    void show intern details ()
    {
        cout <<"\n ---- intern details ---- \n";
        student :: show person details ();
        show employee details ();
    }
};
int main ()
{
    intern it;
    it.student :: get personal details ();
    it.get details ();
    it.get employee details ();
    it.show intern details ();
    return 0;
}
```

PAGE No.
DATE / /

expt. 7

1) WAP using function overloading to calculate the area of a laboratory (rectangular in shape) & area of classroom(square)

ans)
```
#include<iostream>
using namespace std;
class area;
{
    private:
    float l,b;
     public:
    void area (float l);
    {
        float area;
        area = l*b;
        cout <<" area of classroom: " <<area <<endl;
    }
    void area (float l, float b)
    {
        float area;
        area = l*b;
        cout<<" area of laboratory: " <<area;
    }
};
int main()
{
    area a1;
    a1. area (8);
    a1. area (4,12);
    return 0;
}
```

2) WAP using function overloading to calculate the sum of 5 float values & 10 integer values

ans)
```
#include<iostream>
using namespace std;
class sum
{
    private:
    int a,b,c,d,e,f,g,h,i,j;
    float k,l,m,n,o;
    public:
    void sum(float k, float l, float m, float n, float
    {
        float sum;
        sum = k+l+m+n+o;
        cout<<" sum of 5 floating numbers:"<<sum
            <<endl;
    }
    void sum(int a, int b, int c, ind d, int e, int f
            int g, int h, int i, int j)
    {
        int sum;
        sum = a+b+c+d+e+f+g+h+i+j;
        cout<<"sum of 10 integers:"<<sum;
    }
};
int main()
{
    sum s1;
    s1.sum(1.2, 3.5, 5.6, 8.4, 1.5);
    s1.sum(1,2,3,4,5,6,7,8,9,10);
    return 0;
}
```

3) WAP to implement unary operators when used with the object so that the numeric data member of the class is negated

ans)
```
#include<iostream>
using namespace std;
class number
{
    private: int x;
    public void accept()
    {
        cout<<"enter a number:";
        cin>>x;
    void operator -()
    {
        x = -x;
    }
    void display()
    {
        cout<<"negated number:"<<x<<endl;
};
int main()
{
    number n1;
    n1.accept();
    -n1;
    n1.display();
    return 0;
}
```

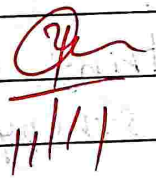
4) WAP to implement the unary ++ operator (for pre~~fi~~ increment & post increment) when used with the object so that the numeric data member of the class is incremented.

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

```
class number
{
    private: int x; int accept;
    public:
    void accept
    {
        cout<<"enter a number";
        cin>>x; temp=x;
    }
    void operator ++()
    {
        x = ++x;
    }
    void reset()
    {
        x=temp;
    }
    void operator ++(int)
    {
        x=x++;
    }
    void display()
    {
        cout<<"(prve) the number is:"<<x<<endl;
    }
    void display2()
    {
        cout<<"(post) the number is:"<<x;
    }
};
int main()
{
```

PAGE No.
DATE / /

```
Number n1;
n1.accept();
++n1;
n1.display();
n1.reset();
n1++;
n1.display2();
return 0;
```



PAGE No.
DATE / /

expt-8

Q) WAP to overload the '+' operator so the two string can be concatinated.

ans)
```
#include <iostream>
#include <string.h>
using namespace std;
class mystring
{
    char str[20];
    public:
    mystring (char, a[20])
    {   strcpy (str, a);
    }
     void display ()
    {
     cout <<" string is : "<<str<<"\n";
    }
     my string operator + (my string);
};
mystring mystring :: operator + (mystring s2);
{
        mystring s3;
        strcpy (s3.str, str);
        strcat (s3. str, s2.str);
        return s3;
}
int main ()
{   mystring s1 ("abc "); mystring s2 ("xyz");
    s3 = s1 + s2;
    cout <<"\n concatinated ";
    s3. display ();
}
```

PAGE No.
DATE / /

2) WAP to create base class I login --:

ans)
```
#include <iostream>
using name space std;
class ilogin
{   protected: string name password;
    public: virtual void accept (){
    cout<<" enter name:"; cin>>name;
    cout<<"password"; cin>>password;
  }
};
class emailogin : public login{
    string email;
    public: void accept () override{
        login :: accept()
        cout<<"enter email:"; cin>>email;
      }
      void display (){
        cout<<"name:" <<name<<"password"
         <<password <<"email" << email<<endl;
       }
};
class membership login: public login {
        string membership;
    public: void accept () override {
    login : accept ();
    cout <<"enter membership ID:";
    cin>>membership ID;
  }
  void display() {
    cout<<In ---membership details---\n";
    cout << name:" name<<"\n password:"<<password
    <<"membership:"<<membership Id <<endl;
  }
```

PAGE No.
DATE / /

};
int main() { email login e; membership login m;
e.accept(); m.accept(); e.display();
m.display();
return 0;
}

PAGE No.
DATE / /

expt- 9

1) WAP to copy the contents of one file into another.

ans)
```
#include <iostream>
#include <fstream>
Using namespace std;
int main(){ fstream first_file;
            fstream second_file;
            first_file.open("first.text",ios::in);
            if(!first_file){
              cout<<"error"<<endl;
              return 1;
            }
           second_file.open("second.text",ios::out);
           if(!second_file){
             cout<<"error"<<endl;
             return 1;
           }
    cout<<"file copied successfully!"<<endl;
    return 0;
}
```

2)

→
```
#include <iostream>
#include <fstream>
#include <cctype>
Using namespace std;
int main(){ fstream new_file;
            new_file.open("first_text",ios::in);
            if(!new_file){
              cout<<"error"<<endl;
              return 1;}
```

```
int digits_count = 0;
int =paces_count=0.
char ch; while (new_file.get(ch)){
    if (isdigit (ch)) {
        digits_count++;
    }
    if(isspace (ch)) {
        space_count++;
    }
new_file.close();
cout<<"No. of digits :"<< digits_count<<
                                    endl
cout<<" No. of spaces :"<<space_count<<
                                    end
    return 0;
}.
```

Q)

→
```
#include <iostream>
#include <fstream>
#include <cctype>
Using namespace std;
fstream new_file
new_file.open ("first.text", ios::in);
(!new_file) {
    cout <<" unter occured !"<< endl;
    return 1;
}
char ch {
    int words_count = 0;
```

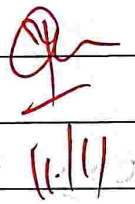
```
bool inWord = false;
while (new_files.get(ch)){
    if (isspace(ch)) {
        inWord = false }
    else if (! inWord)
    {
        words_count++;
        int word = true;
    }
}
};
new_file.close();
cout << "No. of words: " << words_count << endl;
return 0;
}.
```

d)

→

```
#include <iostream>
#include <fstream>
using namespace std;
int main()
{
    ifstream f1("myfile e.txt");
    string ch;
    int w = 0;
    cout << "enter a word:");
    cin >> ch;
    string word;
    while (f1 >> word).
    }
```

```
cout<<"\n"<<word;
if(ch=word)
{ w++
}
}
fl.close()
cout<<"\n occurrance of a word in file i
   <<w; return 0;
}
```



PAGE No.
DATE / /

Experiment. 10

1) WAP to find sum of array using function template

ans)
```
#include<iostream>
using namespace std;
template <class t>
t sum <class t>
t sum (t arr [10])
{     int i;
      t add=0;
      for (i=0; i<10; i++)
      {
          add = add + a[i];
      }
      return add;
}
int main()
{ float arr[] = {12,1,2,5,6,3,4,9,10,14,15};
  cout<<"\n sum of array elements is:"<<sum
  return 0;
}
```

→
```
# include <iostream>
using namespaces std;
template <class T> T square (T num)
{
    return num * num;
}
template <> string square (string str)
{
    return str * str;
} int main (){ int n=5; string t "hello";
```

```
    cout << " square of no. : " << square(n).
    cout << " Square of string : " << square(t);
    return 0;
  }

};
```

→

```
#include <iostream>
Using namespace std;
template <class T> class c
{
  public: T num1=0; T num2=0;
  void accept (){
      cout << " Enter nos :";
      cin>> num1 >> num2;
    }
  void add (){ cout << " sum: " << num1+num2;}
  void sub() { cout << " sub : " << num1-num2; }
  void mult(){ cout << " mult : " << num1 * num2; }
  void div() { cout << " div : " << num1/num2; }
};
int main (){ c <int> obj;
  obj. acc(); obj.add(); obj. sub(); obj. mult();
  obj. div();
  return 0;
}
```

PAGE No.
DATE / /

Experiment No. 11

Q)

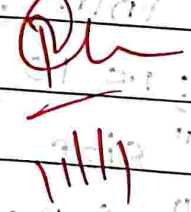
→
```
#include <iostream>
#include <vector>
using namespace std;
int main (){
    vector <char> v(10); unsigned int i;
    cout<<" Size=" <<v.size ()<<endl;
    for(i=0;i<10;i++) { v[i]= i+'a'; }
    cout<<" current content : \n";
    for(i=0;i<10;i++){ v.push_back(i+10+'a'); }
    cout<<" Size
```

again →
```
#include <iostream>
#include <vector>
using namespace std;
int main (){
    vector <char> v(10); unsigned int i;
    cout<<"size="<<v.size ()<<"  "<<endl;
    for(i=0;i<10;i++){ v[i]=i+'a'; }
    cout<<" current content : \n";
    for(i=0;i<v.size ();i++){ cout<<v[i]<<"  "; }
    cout<<" \n\n"; cout<<" expanding vector : \n";
    for(i=0;i<10;i++){ v.push_back (i+10+'a'); }
    cout<<" size now =" <<v.size ()<<endl;
    cout<<" current content : \n";
    for (i=0;i<v.size();i++) { cout<<v[i]<<"  "; }
    cout<<"\n\n";
    for(i=0;i<v.size ();i++){ v[i]=toupper(v[i]); }
    cout<<" modified contents : \n";
    for(i=0;i<v.size();i++) { cout<<v[i]<<"  "; }
```

```
    cout << endl;
    return 0;
}
```

Q)

→
```
#include <iostream>
using namespace std;
int main(){
    vector<char> v(10); vector<char>::iterator p;
    int i; p=v.begin(); i=0;
    while(p != v.end()){
        *p=i+'a'; p++; i++;
    }
    cout<<" original content : \n";
    p=v.begin();
    while( p != v.end()) {
        cout<< *p << " "; p++;
    }
}
```



PAGE No.
DATE / /

Experiment No. 12

Q1)

→
```
#include <bits/stdc++.h>
using namespace std;
int main() {
    stack <char> cars;
    cars.push("BMW"); cars.push("Audi");
    cars.push("Mercedes"); cars.push("Ferrari");
    cout<<"Top element is:" << cars.top()<<endl;
    cout<<"Size of stack is:"<<cars.size()<<endl;
    cars.pop(); cars.pop();
    while(!cars.empty()) {
    cout<<"elements in stack are:"<< cars.top()
                                  <<endl;
    cars.pop();
    }
    return 0;
}
```

→ o/p →
```
Top element : Ferrari
Size of stack is : 4
Elements in stack are : Audi
"  "  "  "  : BMW
```

Q2)

→
```
#include <bits/stdc++.h>
using namespace std;
int main() {
    queue<int> age;
    age.push(21); age.push(22); age.push(23);
    age.push(24);
    cout<<" front is :"<< age.front()<< endl;
    cout<<" Back is :"<< age.back()<< endl;
    age.pop(); age.pop();
    while(!age.empty())
    {
        cout<<" element in queue :"<< age.front()
        <<endl;
        age.pop(); }
    return 0;
}
```

→ O/P → Front element is : 21
Back : 24
Element in queue : 23
Element in queue : 24.