

# **BRCM CET, BAHAL**

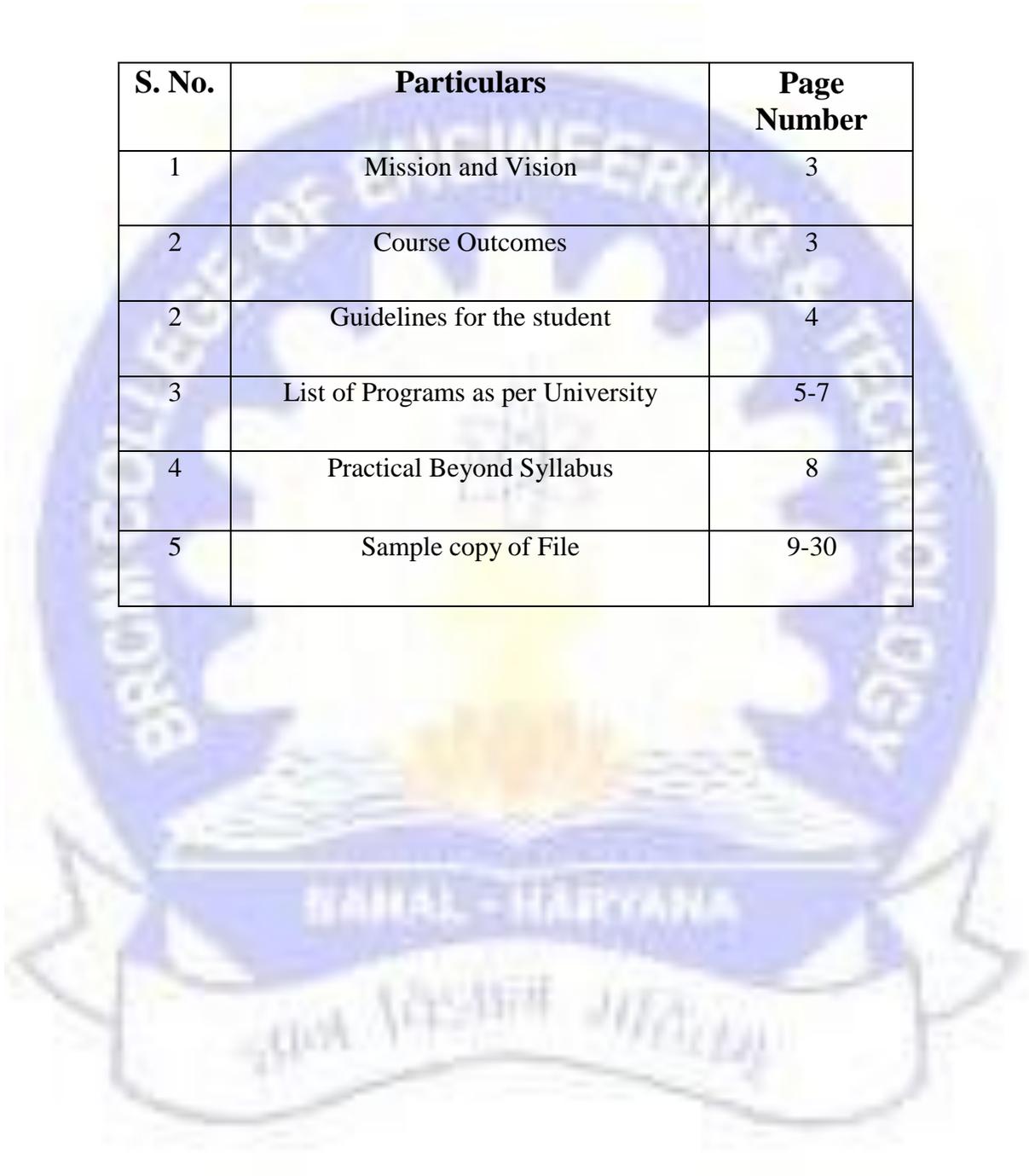


## **LAB MANUAL**

**C++ PROGRAMMING LAB ( LC-CSE-214G)**

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

## Check list for Lab Manual



<b>S. No.</b>	<b>Particulars</b>	<b>Page Number</b>
1	Mission and Vision	3
2	Course Outcomes	3
2	Guidelines for the student	4
3	List of Programs as per University	5-7
4	Practical Beyond Syllabus	8
5	Sample copy of File	9-30

## Department of Computer Science & Engineering

### Vision and Mission of the Department

#### Vision

To be a Model in Quality Education for producing highly talented and globally recognizable students with sound ethics, latest knowledge, and innovative ideas in Computer Science & Engineering.

#### MISSION

To be a Model in Quality Education by

**M1:** Imparting good sound theoretical basis and wide-ranging practical experience to the Students for fulfilling the upcoming needs of the Society in the various fields of Computer Science & Engineering.

**M2:** Offering the Students an overall background suitable for making a Successful career in Industry/Research/Higher Education in India and abroad.

**M3:** Providing opportunity to the Students for Learning beyond Curriculum and improving Communication Skills.

**M4:** Engaging Students in Learning, Understanding and Applying Novel Ideas.

Course: Object Oriented Programming Using C++ LAB

Course Code: LC-CSE-214G

CO (Course Outcomes)		RBT*- Revised Bloom's Taxonomy
CO1	To <b>Explain</b> C++ compiler and translate basic C programs into C++ programs.	L2 (Understand)
CO2	To <b>Apply</b> inheritance and polymorphism features to implement code reusability.	L3 (Apply)
CO3	To <b>Analyze</b> Exception handling concepts to handle runtime errors.	L4 (Analyze)
CO4	To <b>Create</b> programs on classes, objects, constructors and make use of access specified in classes.	L6 (Create)
CO5	To <b>Create</b> programs using different operators, function overloading and operator overloading.	L6 (Create)

### CO PO-PSO Articulation Matrices

Course Outcomes (COs)	(POs)												PSOs	
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1	3	2	2									1	3	1
CO2	2	3	2		2							1	1	1
CO3	3	2	2		2							1	2	2
CO4	3	2	2		2							1	3	3
CO5	2	3	2		2							1	3	3

## **Guidelines for the Students :**

1. Students should be regular and come prepared for the lab practice.
2. In case a student misses a class, it is his/her responsibility to complete that missed experiment(s).
3. Students should bring the observation book, lab journal and lab manual. Prescribed textbook and class notes can be kept ready for reference if required.
4. They should implement the given Program individually.
5. While conducting the experiments students should see that their programs would meet the following criteria:
  - Programs should be interactive with appropriate prompt messages, error messages if any, and descriptive messages for outputs.
  - Programs should perform input validation (Data type, range error, etc.) and give appropriate error messages and suggest corrective actions.
  - Comments should be used to give the statement of the problem and every function should indicate the purpose of the function, inputs and outputs
  - Statements within the program should be properly indented
  - Use meaningful names for variables and functions.
  - Make use of Constants and type definitions wherever needed.
6. Once the experiment(s) get executed, they should show the program and results to the instructors and copy the same in their observation book.
7. Questions for lab tests and exam need not necessarily be limited to the questions in the manual, but could involve some variations and / or combinations of the questions.

## LIST OF PROGRAMS(University Syllabus)

**C++ Programming Lab ( IT 256 E)**

**Semester : IV CSE**

S.NO	PROGRAM
1	<p>Raising a number n to a power p is the same as multiplying n by itself p times. Write a function called power ( ) that takes a double value for n and an int value for p, and returns the result as double value. Use a default argument of 2 for p, so that if this argument is omitted, the number will be squared. Write a main ( ) function that gets values from the user to test this function.</p>
2	<p>A point on the two two numbers can represent dimensional plane: an X coordinate and a Y coordinate. For example, (4,5) represents a point 4 units to the right of the origin along the X axis and 5 units up the Y axis. The sum of two points can be defined as a new point whose X coordinate is the sum of the X coordinates of the points and whose Y coordinate is the sum of their Y coordinates. Write a program that uses a structure called point to model a point. Define three points, and have the user input values to two of them. Then set the third point equal to the sum of the other two, and display the value of the new point. Interaction with the program might look like this:</p> <p>Enter coordinates for P1: 3 4            Enter coordinates for P2: 5 7            Coordinates of P1 + P2 are : 8, 11</p>
3	<p>Create the equivalent of a four function calculator. The program should request the user to enter a number, an operator, and another number. It should then carry out the specified arithmetical operation: adding, subtracting, multiplying, or dividing the two numbers. (It should use a switch statement to select the operation). Finally it should display the result. When it finishes the calculation, the program should ask if the user wants to do another calculation. The response can be 'Y' or 'N'. Some sample interaction with the program might look like this.</p> <p>Enter first number, operator, second number: 10/ 3            Answer = 3.333333            Do another (Y/ N)? Y            Enter first number, operator, second number 12 + 100            Answer = 112            Do another (Y/ N) ? N</p>
4	<p>A phone number, such as (212) 767-8900, can be thought of as having three parts: the area code (212), the exchange (767) and the number (8900). Write a program that uses a structure to store these three parts of a phone number separately. Call the structure phone. Create two structure variables of type phone. Initialize one, and have the user input a number for the other one. Then display both numbers. The interchange might look like this:</p> <p>Enter your area code, exchange, and number: 415 555 1212            My number is (212) 767-8900            Your number is (415) 555-1212</p>

5	<p>Create two classes DM and DB which store the value of distances. DM stores distances in metres and centimeters and DB in feet and inches. Write a program that can read values for the class objects and add one object of DM with another object of DB. Use a friend function to carry out the addition operation.</p> <p>The object that stores the results maybe a DM object or DB object, depending on the units in which the results are required. The display should be in the format of feet and inches or metres and centimetres depending on the object on display.</p>
6	<p>Create a class rational which represents a numerical value by two double values- NUMERATOR &amp; DENOMINATOR. Include the following public member Functions:</p> <ul style="list-style-type: none"> <li>• constructor with no arguments (default).</li> <li>• constructor with two arguments.</li> <li>• void reduce( ) that reduces the rational number by eliminating the highest common factor between the numerator and denominator.</li> <li>• Overload + operator to add two rational number.</li> <li>•</li> <li>• Overload &lt;&lt; operator to enable output through cout.</li> </ul> <p>Write a main ( ) to test all the functions in the class.</p>
7	<p>Consider the following class definition</p> <pre>class father { protected : int age; public; father (int x) {age = x;} virtual void iam ( ) { cout &lt;&lt; "I AM THE FATHER, my age is : "&lt;&lt; age&lt;&lt; endl;} };</pre> <p>Derive the two classes son and daughter from the above class and for each, define iam ( ) to write our similar but appropriate messages. You should also define suitable constructors for these classes. Now, write a main ( ) that creates objects of the three classes and then calls iam ( ) for them. Declare pointer to father. Successively, assign addresses of objects of the two derived classes to this pointer and in each case, call iam ( ) through the pointer to demonstrate polymorphism in action.</p>
8	<p>Write a program that creates a binary file by reading the data for the students from the terminal. The data of each student consist of roll no., name ( a string of 30 or lesser no. of characters) and marks.</p>
9	<p>A hospital wants to create a database regarding its indoor patients. The information to store include</p> <ol style="list-style-type: none"> <li>a) Name of the patient</li> <li>b) Date of admission</li> <li>c) Disease</li> <li>d) Date of discharge</li> </ol> <p>Create a structure to store the date (year, month and date as its members). Create a base class to store the above information. The member function should include functions to enter information and display a list of all the patients in the database. Create a derived class to store the age of the patients. List the information about all the to store the age of the patients. List the information about all the pediatric patients (less than twelve years in age).</p>
10	<p>Make a class Employee with a name and salary. Make a class Manager inherit from</p>

	Employee. Add an instance variable, named department, of type string. Supply a method to toString that prints the manager's name, department and salary. Make a class Executive inherit from Manager. Supply a method to String that prints the string "Executive" followed by the information stored in the Manager superclass object. Supply a test program that tests these classes and methods.
11	Write a function called reversit ( ) that reverses a string (an array of char). Use a for loop that swaps the first and last characters, then the second and next to last characters and so on. The string should be passed to reversit ( ) as an argument. Write a program to exercise reversit ( ). The program should get a string from the user, call reversit ( ), and print out the result. Use an input method that allows embedded blanks. Test the program with Napoleon's famous phrase, "Able was I ere I saw Elba)".
12	Create a base class called shape. Use this class to store two double type values that could be used to compute the area of figures. Derive two specific classes called triangle and rectangle from the base shape. Add to the base class, a member function get_data ( ) to initialize baseclass data members and another member function display_area ( ) to compute and display the area of figures. Make display_area ( ) as a virtual function and redefine this function in the derived classes to suit their requirements. Using these three classes, design a program that will accept dimensions of a triangle or a rectangle interactively and display the area. Remember the two values given as input will be treated as lengths of two sides in the case of rectangles and as base and height in the case of triangles and used as follows: Area of rectangle = x * y Area of triangle = $\frac{1}{2} * x * y$

### Books for Reference :

- C++ How to Program by H M Deitel and P J Deitel, 1998, Prentice Hall
- Object Oriented Programming in Turbo C++ by Robert Lafore ,1994, The WAITE Group Press.
- Programming with C++ By D Ravichandran, 2003, T.M.H
- Object oriented Programming with C++ by E Balagurusamy, 2001, Tata McGraw-Hill
- Computing Concepts with C++ Essentials by Horstmann, 2003, John Wiley,
- The Complete Reference in C++ By Herbert Schildt, 2002, TMH

## Practical beyond Syllabus

P.No.	Program
1.	Write a program to swap two number using the concept of reference variables in the function.
2.	Write a program to implement the concept of static variable and find the Sum of series = $1+2+3+\dots+10$ . Repeat this to find the sum for five times.
3	Write the program to add two Complex number using the concept of Friend function.
4	Write a program to implement the concept of copy constructor.

## Sample copy of file

### PROGRAM NO. 1

```
#include<iostream.h>
#include<conio.h>
void main()
{
    double power(double , int);
    double n,result;
    int p;
    char ch;
    clrscr();
    cout<<"enter the value of number"<<endl;
    cin>>n;
    cout<<"enter the power"<<endl;
    cin>>p;
    result=power(n,p);
    cout<<"result is "<<endl<< result;
    getch();
}
double power(double n,int p)
{
    int i;
    double c=1;
    for(i=0;i<p;i++)
    {
        c=c*n ;
    }
    return(c);
}
```

### PROGRAM NO. 2

```
#include<iostream.h>
#include<conio.h>
class point
{
    int x,y;
    public:
```

```

        void getdata(void);
        void display(void);
        int add(int , int);
    }
    void getdata(void)
    {
        cout<<"enter coordinates "<<endl;
        cin>>x>>y;
    }
    int add(int p , int q)
    {
        int z;
        z=p+q;
        return(z);
    }
void main()
{
    point p1,p2,p3;
    clrscr();
    p3.x=p1.x+p2.x;
    p3.y=p1.y+p2.y;
    cout<<"coordinates of p1+p2 are:";
    cout<<p3.x<<" "<<p3.y;
    getch();
}

```

### **PROGRAM NO. 3**

```

#include<iostream.h>
#include<conio.h>
class operation
{
private:
    int x,y;
    float z;
public:
    void enter();
    void sum();
    void difference();
    void multiply();
    void divide();
};

```

```

void operation::enter()
{
    cout<<"enter first number";
    cin>>x;
    cout<<"enter second number";
    cin>>y;
}
void operation::sum()
{
    z=x+y;
    cout<<"sum is";
    cout<<z;
}
void operation::difference()
{
    z=x-y;
    cout<<"difference is ";
    cout<<z;
}
void operation::multiply()
{
    z=x*y;
    cout<<"Product is ";
    cout<<z;
}
void operation::divide()
{
    z=float(x)/float(y);
    cout<<"result is ";
    cout<<z;
}
void main()
{
    clrscr();
    operation data;
    int a;
    char ch;
    do
    {
        data.enter();
        cout<<"list of operations"<<endl;
        cout<<"1 Addition"<<endl;
        cout<<"2 Subtraction"<<endl;
        cout<<"3 Multiply "<<endl;

```

```

cout<<"4 Divide "<<endl;
cout<<"enter the operation which you want to perform"<<endl;
cin>>a;
switch(a)
{
case 1:
    data.sum();
    break;
case 2:
    data.difference();
    break;
case 3:
    data.multiply();
    break;
case 4:
    data.divide();
    break;
default:
    cout<<"error in choice";
}
cout<<endl<<"want to continue";
cin>>ch;
}
while((ch=='Y')||(ch=='y'));
getch();
}

```

#### **PROGRAM NO. 4**

```

#include<iostream.h>
#include<conio.h>
#include<iomanip.h>
class number
{
private:
    int x,y,z;
public:
    void initial(int a,int b,int c);
    void read();
    void display();
};
void number::initial(int a,int b,int c)

```

```

{
    x=a;
    y=b;
    z=c;
}
void number::read()
{
    cout<<"enter your area code, exchange, and number"<<endl;
    cin>>x>>y>>z;
}
void number::display()
{
    cout<<"("<<x<<")"<<setw(10)<<y<<"-"<<setw(10)<<z;
}
void main()
{
clrscr();
number p1,p2;
p2.read();
p1.initial(212,767,8900);
cout<<endl<<"my number is"<<endl;
p1.display();
cout<<endl<<"your number is "<<endl;
p2.display();
getch();
}

```

### **PROGRAM NO. 5**

```

#include<iostream.h>
#include<conio.h>
#include<iomanip.h>
class db;
class dm
{
private:
    int metre;
    int centi;
public:
    void read()
    {
        cout<<"enter the metres";
        cin>>metre;
    }
}

```

```

    cout<<"enter the centimetres";
    cin>>centi;
}
void display()
{
    cout<< "metres is"<<metre<<"centimetres are"<<centi;
}
void show(dm x)
{
    cout<<setw(5)<<x.metre<<"metre"<<setw(5)<<x.centi<<"centimetres";
}
friend dm add1(dm,db);
friend db add2(dm,db);
};
class db
{
private:
    int feet;
    int inch;
public:
    void read()
    {
        cout<<"enter the feets";
        cin>>feet;
        cout<<"enter the inches";
        cin>>inch;
    }
    void display()
    {
        cout<< "feet is"<<feet<<"inches are"<<inch;
    }
    void show(db x)
    {
        cout<<setw(5)<<x.feet<<" feet"<<setw(5)<< x.inch<<"inches";
    }
    friend db add2(dm,db);
    friend dm add1(dm,db);
};
dm add1(dm x,db y)
{
    dm c1;
    float k;
    int i,m,j,z,n;
    i=y.feet*12;

```

```

    j=i+y.inch;
    k=j*2.54;
    z=int(k)+x.centimeter;
    m=z/100;
    n=z%100;
    c1.metre=x.metre+m;
    c1.centimeter=n;
return(c1);
}
db add2(dm x,db y)
{
db c1;
int a,d,e,c;
float b;
    a=x.metre*100+x.centimeter;
    b=float(a)/2.54;
    c=int(b)+y.inch;
    d=c/12;
    e=c%12;
    c1.inch= e;
    c1.feet=y.feet+d;
return(c1);
}

void main()
{
    int a;
    char ch;clrscr();
    dm d1,d3;
    db d2,d4;
    do
    {
    d1.read();
    d2.read();
    cout<<"Conversion "<<endl<<" 1=Metre Cm" <<endl<<" 2=Feet Inch"<<endl;
    cin>>a;
    switch(a)
    {
    case 1: d3=add1(d1,d2);
            cout<<endl<<"Converted distance into Metre Centimetres "<<endl;
            d3.show(d3);
            break;
    case 2 :d4=add2(d1,d2);
            cout<<endl<<"Converted distance into Feet Inch"<<endl;

```

```

        d4.show(d4);
        break;
    }
    cout<<endl<<"want to continue";
    cin>>ch;
    }
    while((ch=='y')||(ch=='Y'));
    getch();
}

```

---

### **PROGRAM NO. 6**

```

#include<iostream.h>
#include<conio.h>
#include<iomanip.h>
class rational
{
private:
    double num,den;
public:
    rational()           //constructor
    {
        num=1;den=1;
    }
    rational(double a,double b)   //constructor
    {
        num =a;
        den= b;
    }
    void operator>>(rational &s);
    void operator<<(rational &s);
    void display()
    {
        cout<<endl<<"object has values "<<setw(5)<<num<<setw(5)<<den;
    }
    rational operator+(rational &s);
};
void rational ::operator<<(rational &s)
{
    cout<<endl<<"object 1";
    cout<<setw(5)<<num<<setw(5)<<den;
}

```

```

    cout<<endl<<"object 2";
    cout<<setw(5)<<s.num<<setw(5)<<s.den;
    }
void rational::operator>>(rational &s)
    {
    cin>>num>>den>>s.num>>s.den;
    }
rational rational::operator+(rational &s)
    {
    rational temp;
    temp.num=num+s.num;
    temp.den=den+s.den;
    return(temp);
    }
void main()
    {
    clrscr();
    rational o3;           //constructor with no arguments
    o3.display();
    rational o1(2,2);     //constructor with two arguments
    rational o2(5,5);
    o1.display();
    o2.display();
    cout<<endl;
    o1>>o2;               //operator overloading >>
    o1<<o2;               //operator overloading <<
    o3=o1+o2;            //operator overloading +
    o3.display();
    getch();
    }

```

---

### **PROGRAM NO. 7**

```

#include<iostream.h>
#include<conio.h>
class father
{
protected:
    int age;
public:
    father(int x)
    {

```

```

    age=x;
    }
    virtual void iam()
    {
    cout<<"I AM THE FATHER,my age is:"<<age<<endl;
    }
};
class son:public father
{
public:
    son(int x,int y):father(x)
    {
    age=y;
    }
    void iam()
    {
    cout<<"I AM THE SON,my age:"<<age<<endl;
    }
};
class daughter:public father
{
public:
    daughter(int x,int z):father(x)
    {
    age=z;
    }
    void iam()
    {
    cout<<"I AM THE DAUGHTER,my age is:"<<age<<endl;
    }
};
void main()
{
clrscr();
son s(40,10);
daughter d(40,12);
father f(40),*ptr;
ptr=&s;
ptr->iam();
ptr=&d;
ptr->iam();
ptr=&f;
ptr->iam() ;
getch();
}

```

```
}
```

---

## **PROGRAM 8**

```
#include<iostream.h>
#include<fstream.h>
#include<conio.h>
#include<iomanip.h>
class student
{
    char name[30];
    int roll ;
    int marks;
public:
    void input();
    void display();
};
void student::input()
{
    cout<<"enter name";
    cin>>name;
    cout<<" enter roll number";
    cin>>roll;
    cout<<"enter marks ";
    cin>>marks;
}
void student::display()
{
    cout<<setw(15)<<setiosflags(ios::left)<<"\nName :"<<setw(15)<<setiosflags(ios::right)<<name;
    cout<<setw(15)<<setiosflags(ios::left)<<"\nRoll Number:"<<setw(15)<<setiosflags(ios::right)<< roll;
    cout<<setw(15)<<setiosflags(ios::left)<<"\nMarks:"<<setw(15)<<setiosflags(ios::right)<<marks;
}
void main()
{
    clrscr();
    student s[2];
    ofstream outf("record");
    outf.open("record",ios::out);
    for(int i=0;i<2;i++)
    {
        s[i].input();
        outf.write((char*)&s[i],sizeof(s[i]));
```

```

}
outf.close();
cout<<"\n OUTPUT";
ifstream inf("record");
inf.open("record",ios::in);
inf.seekg(0);
for(i=0;i<2;i++)
{
inf.read((char*)&s[i],sizeof(s[i]));
s[i].display();
}
inf.close();
getch();
}

```

### **PROGRAM NO. 9**

```

#include<iostream.h>
#include<conio.h>
#include<fstream.h>
struct date
{
int year;
char month[10];
int dat;
};
class info
{
protected:
char name[20];
struct date x;
char disease[20];
struct date y;
public:
void get();
void display();
};
void info::get()
{
cout<<"enter name of patient"<<endl;
cin>>name;
cout<<"enter date of admission(YY:MM:DD)";
cin>>x.year>>x.month>>x.dat;

```

```

cout<<"enter disease ";
cin>>disease;
cout<<"enter discharge date(MM:YY:DD)";
cin>>y.year>>y.month>>y.dat;
}
void info::display()
{
cout<<"Name: "<<name<<" ...";
cout<<"Date of Admission(MM:YY:DD):";
cout<<x.year<<":"<<x.month<<":"<<x.dat<<endl;
cout<<"Disease :"<<disease<<".... ";
cout<<"Date of discharge(MM:YY:DD):";
cout<<y.year<<":"<<y.month<<":"<<y.dat<<endl;
}
class data:public info
{
    int age;
public:
    void read();
    void show();
    void child();
};
void data::read()
{
cout<<"Enter age";
cin>>age;
}
void data::show()
{
cout<<"Age:"<<age<<endl;
}
void data::child()
{
if(age<12)
{
cout<<endl<<"Name:"<<name<<" ..... "<<"Age:"<<age;
}
}
void main()
{
clrscr();
ofstream outf("record");
data p[1];
outf.open("record",ios::out);

```

```

cout<<"Enter details";
for(int i=0;i<=1;i++)
{
p[i].get();
outf.write((char*)&p[i],sizeof(p[i]));
p[i].read();
outf.write((char*)&p[i],sizeof(p[i]));
}
outf.close();
ifstream inf("record",ios::in);
cout<<"Output : "<<endl;
for(i=0;i<=1;i++)
{
inf.read((char*)&p[i],sizeof(p[i]));
p[i].display();
p[i].show();
}
inf.close();
ifstream in("record",ios::in);
cout<<"List of pediatric patients:";
for(i=0;i<=1;i++)
{
in.read((char*)&p[i],sizeof(p[i]));
p[i].child();
}
getch();
}

```

---

### **PROGRAM NO. 10**

```

#include<iostream.h>
#include<conio.h>
class employee
{
protected:
    char name[20];
    int salary;
public:
    void get()
    {
        cout<<"enter the name of employee"<<endl;
        cin>>name;
    }
}

```

```

        cout<<"enter the salary"<<endl;
        cin>>salary;
    }
};
class manager:public employee
{
protected:
    char department[20];
public:
    void input()
    {
        cout<<"enter the department"<<endl;
        cin>>department;
    }
    void toString()
    {
        cout<<"Name"<<endl;
        cout<<name;
        cout<<endl<<"Department"<<endl;
        cout<<department;
        cout<<endl<<"salary="<<salary;
    }
};
class executive:public manager
{
public:
    void toString()
    {
        cout<<"Executive"<<endl;
    }
};
void main()
{
    executive p;
    clrscr();
    p.get();
    p.input();
    p.toString();
    p.manager::toString();
    getch();
}

```



## **PROGRAM NO. 11**

```
#include<iostream.h>
#include<conio.h>
#include<string.h>
class string
{
private:
    char a[30],temp,i,j,k;
public:
    void input()
    {
        cout<<"enter the string"<<"\n";
        cin.getline(a,30);
    }
    void reverse();
};
void string ::reverse()
{
    k=strlen(a);
    i=0;
    j=k-1;
    while(i<=j)
    {
        temp=a[i];
        a[i]=a[j];
        a[j]=temp;
        i++;
        j--;
    }
    for(i=0;i<k;i++)
    {
        cout<<a[i];
    }
}
void main()
{
    clrscr();
    string p;
    p.input();
    p.reverse();
    getch();
}
```



## **PROGRAM NO. 12**

```
#include<iostream.h>
#include<conio.h>
class shape
{
protected:
    double x,y;
public:
    void get_data()
    {
        cout<<endl<<"enter dimensions";
        cin>>x>>y;
    }
    virtual double display_area()=0;
};
class rectangle:public shape
{
public:
    double display_area()
    {
        return(x*y);
    }
};
class triangle:public shape
{
public:
    double display_area()
    {
        return((x*y)/2);
    }
};
void main()
{
    clrscr();
    shape *p1;
    rectangle r;
    triangle t;
    p1=&r;
    p1->get_data();
    cout<<endl<<"Area of rectangle = ";
    cout<<p1->display_area();
    p1=&t;
    p1->get_data();
```

```
cout<<endl<<"Area of triangle = ";  
cout<<p1->display_area();  
getch();  
}
```

---

