



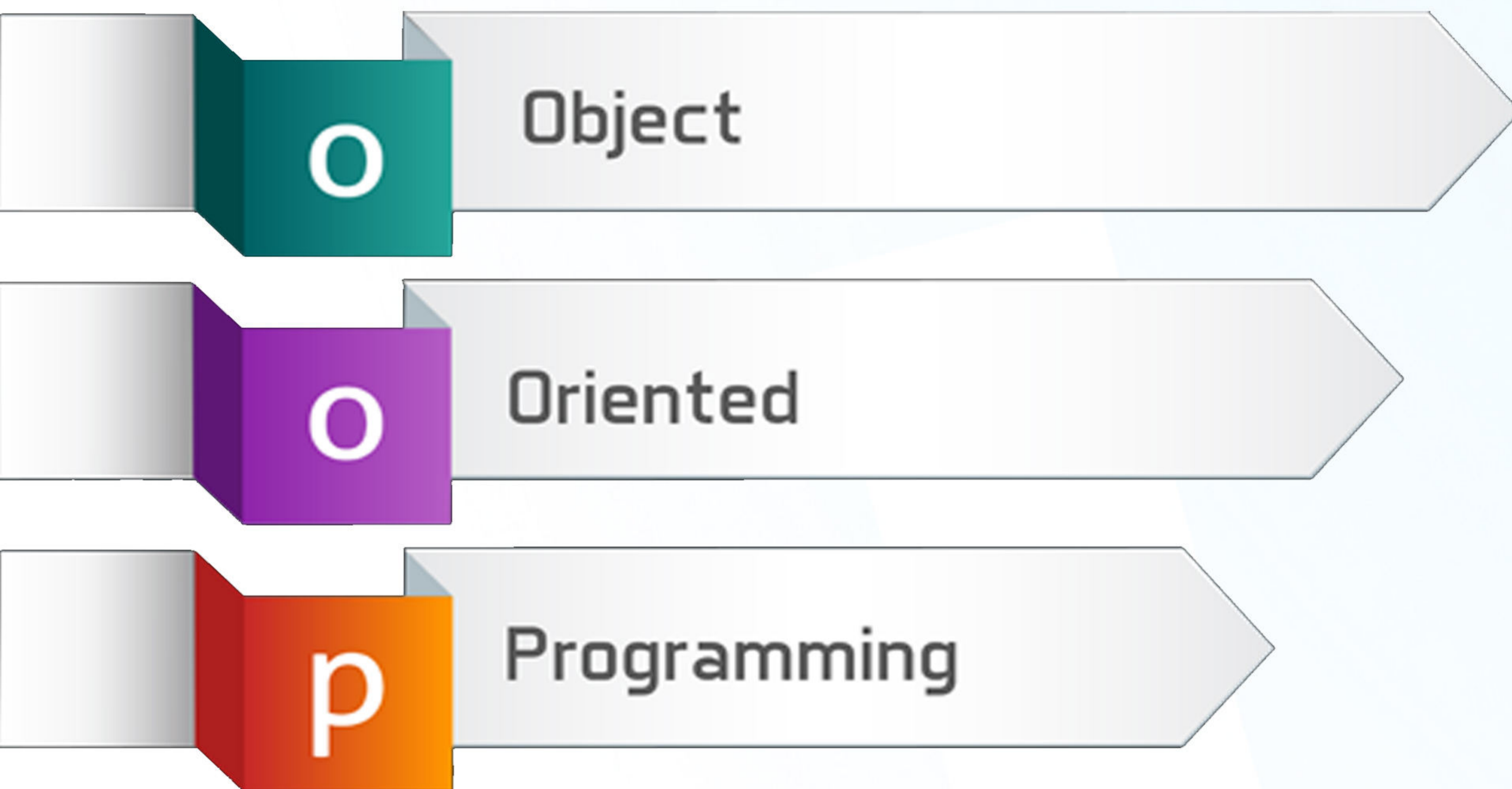
**SIR C.R.R COLLEGE OF ENGINEERING**

**VATLURU - 534007**

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**Object Oriented Programming Lab (CSE 2.1.8)**

**LAB MANUAL**



**Prepared By**

**Y.Leela Sandhya Rani,**

**K.Varada Rajkumar,**

**V.Pranav**



# INDEX

S.No	Name of the Experiment	Page No.
1	Write a C++ Program to evaluate mixed mode expressions and implicit type conversions.	1
2	Write a C++ Program to implement function overloading with various data types, with different number of arguments.	2
3	Write a C++ Program to for performing complex number addition using friend functions.	3
4	Write a C++ Program that implements stack operations using classes and objects.	4
5	Write a C++ Program to implement virtual destructor.	7
6	Write a C++ Program to for performing complex number addition using operator overloading.	8
7	Write a C++ Program to perform string operations using operator overloading.	10
8	Write a C++ Program to implement conversion of objects between different classes using conversion functions.	13
9	Write a C++ Program to show there is ambiguity in multiple inheritance.	15
10	Write a C++ Program on hierarchical inheritance showing public, private and protected inheritances.	16
11	Write a C++ Program for computation of student result using hybrid inheritance.	19
12	Write a C++ Program to implement pointers to derived class and virtual base classes.	21
13	Write a C++ Program to implement virtual functions.	23
14	Write a C++ Program to implement bubble sort using templates.	25
15	Write a C++ Program for handling push on full and pop on empty exceptions for a stack	28
16	Write a C++ Program to implement one Catch Block and all Exceptions	30
17	Write a C++ Program to implement Using Multiple Catch blocks.	31
18	Write a C++ Program for copying one file to another file using streams.	33
19	Write a C++ Program for writing and reading a class object to a file	34
20	Write a C++ Program to mimic a bank management system.	36
21	Write a C++ Program to implement library management system.	39

**1. Write a C++ Program to evaluate mixed mode expressions and implicit type conversions.**

```
#include<iostream.h>
#include<conio.h>
int main()
{
int a,b,c;
float d;
clrscr();
cout<<"Enter Two integer numbers\n";
cin>>a>>b;
cout<<"Enter one float number\n";
cin>>d;
c=a*b*d;
cout<<"Multiplication of float and int into int variable Result is "<<c;
c=d;//converting float to int implicit conversions
cout<<"\nfloat "<<d <<" to int conversion result is "<<c;
char ch = a;
cout<<"\nint "<<a <<" to character result is "<<ch;
return 0;
}
```

**OUTPUT:**

```
Enter two integer numbers
2
3
Enter one float number
6.89
Multiplication of float and int into int variable Result is 41
float 6.89 to int conversion result is 6
int 2 to character result is 8
```

**2. Write a C++ Program to implement function overloading with various data types, with different number of arguments.**

```
#include<iostream.h>
#include<conio.h>

int area(int);
int area(int,int);
int area(int,int,int);
int main()
{
clrscr();
cout<<"\nCalling area() function square(side=5) "<<area(5);
cout<<"\nCalling area() function rectangle(l=5,b=10) "<<area(5,10);
cout<<"\nCalling area() function cuboid(l=5,b=2,h=3) "<<area(5,2,3);
return 0;
}
int area(int a)
{
return(a*a);
}

int area(int l,int b)
{
return(l*b);
}

int area(int l,int b,int h)
{
return(l*b*h);
}
```

**OUTPUT:**

```
Calling area() function square(side=5) 25
Calling area() function rectangle(l=5,b=10) 50
Calling area() function cuboid(l=5,b=2,h=3) 30
```

### 3. Write a C++ Program to for performing complex number addition using friend functions.

```
#include<iostream.h>
class complex
{
int real,imag;
public:
void set()
{
cout<<"enter real and imag part";
cin>>real>>imag;
}
friend complex sum(complex,complex);
void display();
};
void complex::display()
{
cout<<"the sum of complex num is"<<real<<"+i"<<imag;
}
complex sum(complex a,complex b)
{
complex t;
t.real=a.real+b.real;
t.imag=a.imag+b.imag;

return t;
}
int main()
{
complex a,b,c;
a.set();
b.set();
c=sum(a,b);
c.display();
return(0);
}
```

**OUTPUT :**

```
enter real and imag part2 6
enter real and imag part3 7
the sum of complex num is5+i13
```

#### 4. Write a C++ Program that implements stack operations using classes and objects.

```
#include<iostream.h>
#include<conio.h>
#include<stdlib.h>
int top=-1,n;
class stack
{
int a[10];
public:
int getdata();
void push();
void pop();
void display();
};
int stack::getdata()
{
cout<<"enter the size:";
cin>>n;
return n;
}
void stack::push()
{
if(top==n-1)
cout<<"stack is overflow";
else
{
top++;
cout<<"enter a number:";
cin>>a[top];
}
}
void stack::pop()
{
if(top>0)
{
cout<<"deleted element is:"<<a[top];
top--;
}
else
cout<<"stack is underflow";
}
void stack::display()
{
```

```

int i;
if(top==0)
{
cout<<"stack is empty";

}
else
{
cout<<"stack is:";
for(i=0;i<=top;i++)
cout<<a[i]<<"\t";
}
}
int main()
{
stack s;
int ch,n;
clrscr();
n=s.getdata();
if(n>0)
{
do
{
cout<<"\n1.push\t2.pop\t3.display\t4.exit\t";
cout<<"\nenter your choice:";
cin>>ch;
switch(ch)
{
case 1:
s.push();

break;
case 2:
s.pop();
break;
case 3:
s.display();
break;
case 4:
exit(1);
default:cout<<"invalid choice";
}
}while(1);
}
return 0;
}

```

**OUTPUT:**

```
enter the size:3

1.push  2.pop   3.display      4.exit
enter your choice:1
enter a number:111

1.push  2.pop   3.display      4.exit
enter your choice:1
enter a number:222

1.push  2.pop   3.display      4.exit
enter your choice:1
enter a number:333

1.push  2.pop   3.display      4.exit
enter your choice:3
stack is:111    222    333
1.push  2.pop   3.display      4.exit
enter your choice:2
deleted element is:333
1.push  2.pop   3.display      4.exit
enter your choice:3
stack is:111    222
1.push  2.pop   3.display      4.exit
enter your choice:4_
```



## 5. Write a C++ Program to implement virtual destructor.

```
#include<iostream.h>
#include<conio.h>
class Base
{
public:
virtual ~Base() {cout << "Base Destructor \n"; }
};
class Derived:public Base
{
public:
~Derived() { cout<< "Derived Destructor"; }
};
int main()
{
Base* b = new Derived; //Upcasting
clrscr();
delete b;
return 0;
}
```

### OUTPUT:



```
Derived Destructor
Base Destructor
```

**6. Write a C++ Program to for performing complex number addition using operator overloading.**

```
#include<iostream.h>
class complex
{
float x;
float y;
public:
complex()
{}
complex(float real,float imag)
{
x=real;
y=imag;
}
complex operator+(complex);
void display();
};
complex complex:: operator+(complex c)
{
complex temp;
temp.x=x+c.x;
temp.y=y+c.y;
return temp;

}
void complex::display()
{
cout<<x<<"+"<<y<<"\n";
}
int main()
{
complex C1,C2,C3;
float x1,y1,x2,y2;
cout<<"enter the value of x1:";
cin>>x1;
cout<<"enter the value of y1:";
cin>>y1;
C1=complex(x1,y1);
cout<<"enter the value of x2:";
cin>>x2;
cout<<"enter the value of y2:";
cin>>y2;
C2=complex(x2,y2);
```

```
C3=C1+C2;
cout<<"C1=";
C1.display();
cout<<"C2=";
C2.display();

cout<<"C3=";
C3.display();
return 0;
}
```

**OUTPUT:**

```
enter the value of x1:4
enter the value of y1:5
enter the value of x2:2
enter the value of y2:7
C1=4+j5
C2=2+j7
C3=6+j12
```

## 7. Write a C++ Program to perform string operations using operator overloading.

```
#include<iostream.h>
#include<string.h>
#include<stdio.h>
#include<conio.h>
#define max 50

class string
{
char str[max];
public:
friend void operator>>(istream &din,string &s1)
{
din>>s1.str;
}
friend void operator<<(ostream &dout,string &s1)
{
dout<<s1.str;
}
void operator+=(string s1);
void operator==(string s1);
void operator=(string s1);
string()
{

strcpy(str,"");
}
string(char s[])
{
strcpy(str,s);
}
};
void string::operator+=(string s1)
{
if(strlen(str)+strlen(s1.str)<max)
{
strcat(str,s1.str);
cout<<"\nconcatenated string is "<<str;
}
else
{
cout<<"\nno space to add";
}
}
void string::operator==(string s1)
```



```

{
if(strcmp(str,s1.str)==0)
{
cout<<"\nstrings are equal";
}

else
{
cout<<"\n strings are not equal";
}
}
void string::operator=(string s1)
{
strcpy(str,s1.str);
cout<<"\ncopied string is"<<str;
}
int main()
{
int ch;
char ans;
string s1,s2;
clrscr();
cout<<"*****STRING OPERATIONS*****";
do
{
cout<<"\nyour choices\n\t\t1.concatenation\n\t\t2.comparision\n\t\t3.copy";
cout<<"\nenter choice";
cin>>ch;
switch(ch)
{
case 1:cout<<"enter string1:";

cin>>s1;
cout<<"enter string2:";
cin>>s2;
s1+=s2;
break;
case 2:cout<<"enter string 1";
cin>>s1;
cout<<"enter string2";
cin>>s2;
s1==s2;
break;
case 3:
cout<<"enter string to be copied";
cin>>s1;

```

```

s2=s1;
break;
default:
cout<<"wrong choice";
break;
}
cout<<"\ndo you want to continue(yes/no)=(y/n) ";
cin>>ans;
}while(ans=='y' || ans=='Y');
return 0;
}

```

### OUTPUT:

```

*****STRING OPERATIONS*****
your choices
                1.concatenation
                2.comparison
                3.copy
enter choice1
enter string1:CRR
enter string2:COE

concatenated string is CRRCOE
do you want to continue(yes/no)=(y/n) Y

your choices
                1.concatenation
                2.comparison
                3.copy
enter choice2
enter string 1CSE
enter string2CSE

strings are equal
do you want to continue(yes/no)=(y/n) Y
enter choice3
enter string to be copiedCRRCSE

copied string isCRRCSE
do you want to continue(yes/no)=(y/n) N

```

**8. Write a C++ Program to implement conversion of objects between different classes using conversion functions.**

```
#include <iostream.h>
#include <conio.h>
class invent2;
class invent1
{
    int code;
    int items;
    float price;
public:
    invent1(int a, int b,int c)
    {
        code=a;
        items=b;
        price=c;
    }
    void putdata()
    {
        cout<<" Code: "<<<code <<"\n";
        cout<<" Items: "<<<items <<"\n";
        cout<<" Price: "<<<price <<"\n";
    }

    int getcode(){return code;}
    int getitems(){return items;}
    float getprice(){return price;}
    operator float(){return(items*price);}
};
class invent2
{
    int code;
    float value;
public:
    invent2()
    {
        code=0;
        value=0;
    }
    invent2(int x,float y)
    {
        code=x;
        value=y;
    }
}
```

```

void putdata()
{
cout<<" Code: "<<code <<"\n";
cout<<" Value: "<<value <<"\n\n";

}
invent2(invent1 p)
{
code=p.getcode();
value=p.getititems()*p.getprice();
}
};

int main()
{
invent1 s1(100,5,140.0);
invent2 d1;
float total;
clrscr();
total=s1;
d1=s1;
cout<<"Product details - invent1 type \n";
s1.putdata();
cout<<"\n Stock Value \n";
cout<<"Value "<<total<<"\n\n";
cout<<"Product details - invent2 type \n";
d1.putdata();
return 0;
}

```

## OUTPUT:

```

Product details - invent1 type
Code: 100
Items: 5
Price: 140

Stock Value
Value 700

Product details - invent2 type
Code: 100
Value: 700

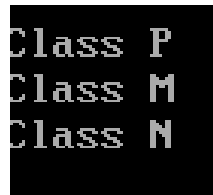
```



**9. Write a C++ Program to show there is ambiguity in multiple inheritance.**

```
#include<iostream.h>
#include<conio.h>
class M
{
public:
void display()
{
cout<<"Class M\n";
}
};
class N
{
public:
void display()
{
cout<<"Class N\n";
}
};
class P : public M,public N
{
public:
void display1()
{
cout<<"Class P\n";
M:: display();
N:: display();
// display();ambiguity error
}
};
int main()
{
P p;
clrscr();
p.display1();
return 0;
}
```

**OUTPUT:**



```
Class P
Class M
Class N
```

## 10. Write a c++ program to implement hierarchical inheritance

```
#include<iostream.h>
#include<conio.h>
class A
{
    int x;
protected:
    int y;
public:
    int z;
    void set_xyz()
    {
        cout<<"Enter Three values\n";
        cin>>x>>y>>z;
    }

    void put_xyz()
    {
        cout<<"X="<<x<<" Y="<<y<<" Z="<<z;
    }
};

class B : public A
{
public:
    void setdataB()
    {
        cout<<"\nPublic derivation\n";
        cout<<"\nPublic and Protected data are as same public and protected in derived class";
        cout<<"\nEnter Y and Z values\n";
        cin>>y>>z;
    }
    void putdataB()
    {
        cout<<"\nY="<<y;
        cout<<"\nZ="<<z;
    }
};
class C :protected A
{
public:
    void setdataC()
    {
        cout<<"\nProtected derivation\n";
```

```

cout<<"\nPublic and Protected data are protected in derived class";
cout<<"\nEnter Y and Z values\n";
cin>>y>>z;
}

void putdataC()
{
cout<<"\nY="<<y;
cout<<"\nZ="<<z;
}
};
class D : private A
{
public :
void setdataD()
{
cout<<"\nPrivate derivation\n";
cout<<"\nPublic and Protected data are private in derived class";
cout<<"\nEnter Y and Z values\n";
cin>>y>>z;
}
void putdataD()
{
cout<<"\nY="<<y;
cout<<"\nZ="<<z;
}
};

int main()
{
A a;
clrscr();
a.set_xyz();
a.put_xyz();
cout<<"\n a.z="<<a.z;

B b;
b.setdataB();
b.putdataB();
cout<<"\nb.z="<<b.z;

C c;
c.setdataC();
c.putdataC();
//cout<<c.z; not accessible

```

```
D d;  
d.setdataD();  
d.putdataD();  
//cout<<d.z; not accessible  
return 0;  
}
```

## OUTPUT:

```
Enter Three values  
11  
23  
45  
X=11 Y=23 Z=45  
a.z=45  
Public derivation  
  
Public and Protected data are as same public and protected in derived class  
Enter Y and Z values  
44  
55  
  
Y=44  
Z=55  
b.z=55  
Protected derivation  
  
Public and Protected data are protected in derived class  
Enter Y and Z values
```



## 11. Write a C++ Program for computation of student result using hybrid inheritance.

```
#include<iostream.h>
class student
{
protected:
int rno;
public:
void getno(int a)
{
rno=a;
}
void putno(void)
{
cout<<"rollno:"<<rno<<"\n";
}
};
class test:public student
{
protected:
float part1,part2;
public:
void getmarks(float x,float y)
{
part1=x;
part2=y;
}
void putmarks(void)
{
cout<<"marks obtained:"<<"\n"<<"part1="<<part1<<"\n"<<"part2="<<part2<<"\n";
}
};
class sports
{
protected:
float score;
public:
void getscore(float s)
{
score=s;
}
void putscore(void)
{
cout<<"sports wt:"<<score<<"\n\n";
}
}
```

```

};
class result:public test,public sports
{
float total;
public:
void display(void);
};
void result::display(void)
{
total=part1+part2+score;
putno();
putmarks();
putscore();
cout<<"total score:"<<total<<"\n";
}
int main()
{
result student;
student.getno(1234);
student.getmarks(45.5,35);
student.getscore(6.0);
student.display();
return 0;
}

```

### OUTPUT :

```

rollno:1234
marks obtained:
part1=45.5
part2=35
sports wt:6

total score:86.5
rollno:1234
marks obtained:
part1=45.5
part2=35
sports wt:6

total score:86.5

```

**12. Write a C++ Program to implement pointers to derived class and virtual base classes.**

```
#include<iostream.h>
#include<conio.h>
class BC
{
public:
    int b;
    void show()
    {
        cout<<"b= "<<b<<"\n";
    }
};

class DC : public BC
{
public:
    int d;
    void show()
    {
        cout<<"b= "<<b<<"\n";
        cout<<"d= "<<d<<"\n";
    }
};

int main()
{
    BC *bptr;
    BC base;
    clrscr();
    bptr=&base;

    bptr->b=100;
    cout<<"bptr points to base object \n";
    bptr->show();
    //Derived class
    DC derived;
    bptr=&derived;
    bptr->b=200;
    /* bptr->d=300 ;*/ // won't work
    cout<<"bptr points to derived object \n";
    bptr->show();

    DC *dptr;
    dptr=&derived;
```

```
dptr->d=300;

cout<<"dptr is derived type pointer \n";
dptr->show();

return 0;
}
```

### OUTPUT:

```
bptr points to base object
b= 100
bptr points to derived object
b= 200
dptr is derived type pointer
b= 200
d= 300
```

### 13. Write a C++ Program to implement virtual functions.

```
#include<iostream.h>
#include<conio.h>
class base
{
    public:
        virtual void show()
        {
            cout<<"\n Base class show:";
        }
        void display()
        {
            cout<<"\n Base class display:" ;
        }
};

class drive:public base
{
    public:
        void display()
        {
            cout<<"\n Drive class display:";
        }
        void show()
        {
            cout<<"\n Drive class show:";
        }
};

void main()
{
    clrscr();
    base obj1;
    base *p;
    cout<<"\n\t P points to base:\n" ;

    p=&obj1;
    p->display();
    p->show();

    cout<<"\n\n\t P points to drive:\n";
    drive obj2;
    p=&obj2;
    p->display();
}
```

```
p->show();  
getch();  
}
```

**OUTPUT:**

```
      P points to base:  
  
Base class display:  
Base class show:  
  
      P points to drive:  
  
Base class display:  
Drive class show:
```

#### 14. Write a C++ Program to implement bubble sort using templates.

```
#include<iostream.h>
#include<iomanip.h>
#include<stdlib.h>
template <class t>
class bubble
{
t a[10];

public:
void get(int);
void sort(int);
void display(int);
};
template <class t>
void bubble <t>::get(int n)
{
int i;
cout<<"\nenter n elements";
for(i=0;i<n;i++)
cin>>a[i];
}
template <class t>
void bubble <t>::sort(int n)
{
t temp;
int i,j;
for(i=0;i<n;i++)
for(j=i+1;j<n;j++)
if(a[i]>a[j])
{
temp=a[i];
a[i]=a[j];
a[j]=temp;
}
}
template <class t>
void bubble <t>::display(int n)
{
int i;
cout<<"\nthe sorted array is";
for(i=0;i<n;i++)
cout<<a[i]<<setw(10);
}
}
```

```

int main()
{
int n,ch;
while(1)
{
cout<<"\nbubble sort choices:\n";
cout<<"\n1.int\t2.float\n3.char\t4.exit";
cout<<"\nenter your choice";
cin>>ch;
switch(ch)
{
case 1:{
cout<<"\nINTEGER SORTING";
bubble <int> b1;
cout<<"\nenter no of elements";
cin>>n;
b1.get(n);
b1.sort(n);
b1.display(n);
break;
};
case 2:{
cout<<"\nFLOAT SORTING";
bubble <float> b2;
cout<<"\nenter no of elements";
cin>>n;
b2.get(n);
b2.sort(n);
b2.display(n);
break;
};
case 3:{
cout<<"\nCHARACTER SORTING";
bubble <char> b3;
cout<<"\nenter no of elements";
cin>>n;
b3.get(n);
b3.sort(n);
b3.display(n);
break;
};
case 4:{
cout<<"\nterminating sorting";
exit(0);
};
default:cout<<"\nwrong choice";

```



```
break;
}
}
}
```

## OUTPUT:

```
bubble sort choices:
1.int  2.float
3.char 4.exit
enter your choice1

INTEGER SORTING
enter no of elements4
enter n elements44 33 22 11
the sorted array is11      22      33      44
bubble sort choices:
1.int  2.float
3.char 4.exit
enter your choice2

FLOAT SORTING
enter no of elements3
enter n elements5.6 15.6 2.3
the sorted array is2.3      5.6      15.6
bubble sort choices:
1.int  2.float
3.char 4.exit
enter your choice3

CHARACTER SORTING
enter no of elements4
enter n elementsz v a b
the sorted array isabvz
bubble sort choices:
1.int  2.float
3.char 4.exit
enter your choice4_
```

**15. Write a C++ Program for handling push on full and pop on empty exceptions for a stack.**

```
#include<iostream>
using namespace std;
int main()
{
int i,ch,no,count=0,arr[3];
cout<<"\n1.push\n2.pop\n3.showstack\n4.exit\n";
try
{
while(1)
{
cout<<"\nenter your choice:";
cin>>ch;
if(ch==1)
{
if(count==3)
throw 1;
cout<<"enter no to be pushed:";
cin>>no;
arr[count]=no;
count++;
}
else if(ch==2)
{
if(count==0)
throw 'a';
count--;
cout<<"the element deleted is:"<<arr[count]<<endl;
}
else if(ch==3)
{
if(count==0)
cout<<"stack is empty";
else
{
cout<<"stack contains";
for(i=0;i<count;i++)
cout<<arr[i]<<"\t";
cout<<endl;
}
}
else if(ch==4)
exit(0);
}
```

```
else
cout<<"wrong choice";
}
}
catch(char c)
{
cout<<"caught a pop on empty exception";
}
catch(int a)
{
cout<<"caught a push on full exception";
}
return 0;
}
```

### **OUTPUT:**

```
1.push
2.pop
3.showstack
4.exit
enter your choice 1
enter no to be pushed 23
1.push
2.pop
3.showstack
4.exit
enter your choice 1
enter no to be pushed 45
1.push
2.pop
3.showstack
4.exit
enter your choice 1
enter no to be pushed 67
1.push
2.pop
3.showstack
4.exit
enter your choice 1
caught a push on full exception
```

## 16. Write a C++ Program to implement One Catch Block and all Exceptions

```
#include<iostream>
using namespace std;
void test(int x)
{
    try
    {
        if (x==0) throw x;
        if (x==-1) throw 'x';
        if (x==1)throw 1.0;
    }
    catch(...)
    {
        cout<<"Caught an exception\n";
    }
}
int main()
{
    cout<<"Testing catch all exception\n";
    test(-1);
    test(0);
    test(1);
    return 0;
}
```

### OUTPUT:

```
Testing catch all exception
Caught an exception
Caught an exception
Caught an exception
```

## 17. Write a C++ program to implement Multiple Catch blocks.

```
#include<iostream>
using namespace std;
void test(int x)
{
    try {
        if(x==1) throw x;
        else if(x==0) throw 'x';
        else if(x==-1) throw 1.0;
        cout<<"End of try-block \n";
    }

    catch(char c)
    {
        cout<<"Caught a character \n";
    }
    catch(int m)
    {
        cout<<"Caught an integer \n";
    }
    catch(double d)
    {
        cout<<"Caught a double \n";
    }
    cout<<"End of try-catch system \n\n";
}
int main()
{
    cout<<"Testing multiple catches \n";
    cout<<"x=1 \n";
    test(1);
    cout<<"x=0 \n";
    test(0);
    cout<<"x=-1 \n";
    test(-1);
    cout<<"x=2 \n";
    test(2);
    return 0;
}
```

## **OUTPUT:**

Testing multiple catches

x=1

caught an integer

End of try-catch system

x=0

caught a character

End of try-catch system

x=-1

caught a double

End of try-catch system

x=2

End of try-block

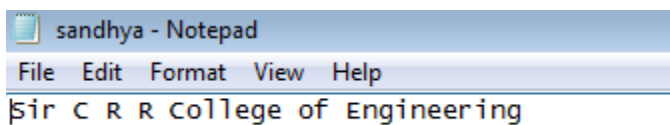
End of try-catch system

## 18. Write a C++ Program for copying one file to another file using streams.

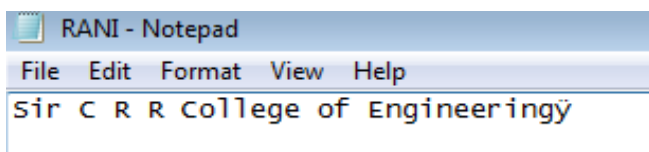
```
#include<iostream>
#include<fstream>
using namespace std;
int main()
{
char s[60],t[60];
char ch;
cout<<endl<<"enter source filename:";
cin>>s;
cout<<endl<<"enter target filename:";
cin>>t;
ifstream infile(s);
ofstream outfile(t);
while(infile)
{
infile.get(ch);
outfile.put(ch);
}
return 0;
}
```

### OUTPUT:

```
enter source filename:sandhya.txt
enter target filename:rani_
```



sandhya - Notepad  
File Edit Format View Help  
sir C R R College of Engineering



RANI - Notepad  
File Edit Format View Help  
sir C R R College of Engineering

## 19. Write a C++ Program for writing and reading a class object to a file.

```
#include<iostream>
#include<fstream>
#include<iomanip>
using namespace std;
class INVENTORY
{
char name[10];
int code;
float cost;
public:
void readdata(void);
void writedata(void);
};
void INVENTORY::readdata(void)
{
cout<<"enter name:";
cin>>name;
cout<<"enter code:";
cin>>code;
cout<<"enter cost:";
cin>>cost;
}
void INVENTORY::writedata(void)
{
cout<<setw(10)<<name
<<setiosflags(ios::right)
<<setw(10)<<code
<<setprecision(5)
<<setw(10)<<cost
<<endl;
}
int main()
{
INVENTORY item[3];
fstream file;
file.open("STOCK.DAT",ios::out);
cout<<"enter details for three items\n";
for( i=0;i<3;i++)
{
item[i].readdata();
file.write((char *)&item[i],sizeof(item[i]));
}
file.seekg(0);
cout<<"\noutput\n\n";
```



```
for(int i=0;i<3;i++)
{
file.read((char *)&item[i],sizeof(item[i]));
item[i].writedata();
}
file.close();
return 0;
}
```

### OUTPUT:

```
enter details for three items
enter name:sandhya
enter code:2314
enter cost:567
enter name:madhu
enter code:2315
enter cost:562
enter name:uday
enter code:2316
enter cost:677

output

    sandhya      2314      567
    madhu       2315      562
    uday        2316      677
```

## 20. Write a C++ Program to mimic a bank management system

*Program to implement a class to represent a bank account. Include the following*

### *Data members*

- a) *Name of the depositor*
- b) *Account Number*
- c) *Type of Account*
- d) *Balance amount in the account*

### *Member functions*

- a) *To assign initial values*
- b) *To deposit an amount*
- c) *To withdraw an amount after checking the balance*
- d) *To display name and balance*

*Write a main program to test the program.*

```
#include<iostream.h>
class bank
{
    char name[30];
    char account_number[10];
    char account_type[10];
    float bal;
public:
    void initial()
    {
        cout<<"\nenter the name of the depositer: ";
        cin>>name;
        cout<<"\nenter the account number of the depositer: ";
        cin>>account_number;
        cout<<"\nenter the type of the account: ";
        cin>>account_type;
        cout<<"\nenter the balance: ";
        cin>>bal;
    }
    void deposit()
    {
        float amt;
        cout<<"enter the amount to be deposited: ";
        cin>>amt;
        bal=bal+amt;
        cout<<"\nthe balance after deposited "<<bal;
    }
    void withdraw()
    {
        float amt;
```

```

cout<<"\nenter tyhe amount to be withdrawn: ";
cin>>amt;
if(bal-amt<500)
{
    cout<<"\nwithdraw is not possible: ";
}
else
    bal=bal-amt;
}
void display()
{
    cout<<"\nname: "<<name;
    cout<<"\nbalance: "<<bal;
}
};
int main()
{
    bank ob1;
    int n;
    cout<<"\nenter the details of the customer:\n";
    ob1.initial();
    while(1)
    {
        cout<<"\n1.deposit\n2.withdraw\n3.display\n4.exit\n";
        cout<<"\nenter your choice";
        cin>>n;
        switch(n)
        {
            case 1:ob1.deposit();
                break;
            case 2:ob1.withdraw();
                break;
            case 3:ob1.display();
                break;
            case 4:exit(0);
        }
    }
}
}

```

**Output:-**

enter the details of the customer:

enter the name of the depositer: Pranav

enter the account number of the depositer: 123456

enter the type of the account: ac

enter the balance: 2000

- 1.deposit
- 2.withdraw
- 3.display
- 4.exit

enter your choice1

enter the amount to be deposited: 200

the balance after deposited: 2200

- 1.deposit
- 2.withdraw
- 3.display
- 4.exit

enter your choice2

enter the amount to be withdrawn: 3000

withdraw is not possible:

- 1.deposit
- 2.withdraw
- 3.display
- 4.exit

Enter your choice 4

## 21. Write a C++ Program to implement library management system.

LIBRARY MANAGEMENT SYSTEM has **book and student class** with data members like book no, bookname, authername. Books records is stored in a **binary file**. A student can issue book and deposit it within 15 days. Student is allowed to issue only one book. Student Records are stored in binary file. Administrator can add, modify or delete record.

Source Code:

```
//*****  
//          HEADER FILE USED IN PROJECT  
//*****  
  
#include<fstream.h>  
#include<conio.h>  
#include<stdio.h>  
#include<process.h>  
#include<string.h>  
#include<iomanip.h>  
  
//*****  
//          CLASS USED IN PROJECT  
//*****  
  
class book  
{  
    char bno[6];  
    char bname[50];  
    char aname[20];  
public:  
    void create_book()  
    {  
        cout<<"\nNEW BOOK ENTRY...\n";  
        cout<<"\nEnter The book no.";  
        cin>>bno;  
        cout<<"\n\nEnter The Name of The Book ";  
        gets(bname);  
        cout<<"\n\nEnter The Author's Name ";  
        gets(aname);  
        cout<<"\n\nBook Created..";  
    }  
  
    void show_book()
```

```

    {
        cout<<"\nBook no. : "<<bno;
        cout<<"\nBook Name : ";
        puts(bname);
        cout<<"Author Name : ";
        puts(aname);
    }

void modify_book()
{
    cout<<"\nBook no. : "<<bno;
    cout<<"\nModify Book Name : ";
    gets(bname);
    cout<<"\nModify Author's Name of Book : ";
    gets(aname);
}

char* retbno()
{
    return bno;
}

void report()
{cout<<bno<<setw(30)<<bname<<setw(30)<<aname<<endl;}

}; //class ends here

```

```

class student
{
    char admno[6];
    char name[20];
    char stbno[6];
    int token;
public:
    void create_student()
    {
        clrscr();
        cout<<"\nNEW STUDENT ENTRY...\n";
        cout<<"\nEnter The admission no. ";
        cin>>admno;
        cout<<"\nEnter The Name of The Student ";
        gets(name);
        token=0;
        stbno[0]='\0';
    }
}

```

```

        cout<<"\n\nStudent Record Created..";
    }

void show_student()
{
    cout<<"\nAdmission no. : "<<admno;
    cout<<"\nStudent Name : ";
    puts(name);
    cout<<"\nNo of Book issued : "<<token;
    if(token==1)
        cout<<"\nBook No "<<stbno;
}

void modify_student()
{
    cout<<"\nAdmission no. : "<<admno;
    cout<<"\nModify Student Name : ";
    gets(name);
}

char* retadmno()
{
    return admno;
}

char* retstbno()
{
    return stbno;
}

int rettoken()
{
    return token;
}

void addtoken()
{token=1;}

void resettoken()
{token=0;}

void getstbno(char t[])
{
    strcpy(stbno,t);
}

void report()
{cout<<"\t"<<admno<<setw(20)<<name<<setw(10)<<token<<endl;}

```

```
}; //class ends here
```

```
/**
//      global declaration for stream object, object
**
```

```
fstream fp,fp1;
book bk;
student st;
```

```
/**
//      function to write in file
**
```

```
void write_book()
{
    char ch;
    fp.open("book.dat",ios::out|ios::app);
    do
    {
        clrscr();
        bk.create_book();
        fp.write((char*)&bk,sizeof(book));
        cout<<"\n\nDo you want to add more record..(y/n?)";
        cin>>ch;
    }while(ch=='y' || ch=='Y');
    fp.close();
}
```

```
void write_student()
{
    char ch;
    fp.open("student.dat",ios::out|ios::app);
    do
    {
        st.create_student();
        fp.write((char*)&st,sizeof(student));
        cout<<"\n\nDo you want to add more record..(y/n?)";
        cin>>ch;
    }while(ch=='y' || ch=='Y');
    fp.close();
}
```



```

//*****
//      function to read specific record from file
//*****

void display_spb(char n[])
{
    cout<<"\nBOOK DETAILS\n";
    int flag=0;
    fp.open("book.dat",ios::in);
    while(fp.read((char*)&bk,sizeof(book)))
    {
        if(strcmpi(bk.retbn(),n)==0)
        {
            bk.show_book();
            flag=1;
        }
    }

    fp.close();
    if(flag==0)
        cout<<"\n\nBook does not exist";
    getch();
}

void display_sps(char n[])
{
    cout<<"\nSTUDENT DETAILS\n";
    int flag=0;
    fp.open("student.dat",ios::in);
    while(fp.read((char*)&st,sizeof(student)))
    {
        if((strcmpi(st.retadmno(),n)==0))
        {
            st.show_student();
            flag=1;
        }
    }

    fp.close();
    if(flag==0)
        cout<<"\n\nStudent does not exist";
    getch();
}

//*****

```

```
//      function to modify record of file
//*****
```

```
void modify_book()
{
    char n[6];
    int found=0;
    clrscr();
    cout<<"\n\n\tMODIFY BOOK REOCORD.... ";
    cout<<"\n\n\tEnter The book no. of The book";
    cin>>n;
    fp.open("book.dat",ios::in|ios::out);
    while(fp.read((char*)&bk,sizeof(book)) && found==0)
    {
        if(strcmpi(bk.retbn(),n)==0)
        {
            bk.show_book();
            cout<<"\nEnter The New Details of book"<<endl;
            bk.modify_book();
            int pos=-1*sizeof(bk);
            fp.seekp(pos,ios::cur);
            fp.write((char*)&bk,sizeof(book));
            cout<<"\n\n\t Record Updated";
            found=1;
        }
    }

    fp.close();
    if(found==0)
        cout<<"\n\n Record Not Found ";
    getch();
}
```

```
void modify_student()
{
    char n[6];
    int found=0;
    clrscr();
    cout<<"\n\n\tMODIFY STUDENT RECORD... ";
    cout<<"\n\n\tEnter The admission no. of The student";
    cin>>n;
    fp.open("student.dat",ios::in|ios::out);
    while(fp.read((char*)&st,sizeof(student)) && found==0)
    {
        if(strcmpi(st.retadmno(),n)==0)
        {
```

```

        st.show_student();
        cout<<"\n\nEnter The New Details of student"<<endl;
        st.modify_student();
        int pos=-1*sizeof(st);
        fp.seekp(pos,ios::cur);
        fp.write((char*)&st,sizeof(student));
        cout<<"\n\n\t Record Updated";
        found=1;
    }
}

fp.close();
if(found==0)
    cout<<"\n\n Record Not Found ";
getch();
}

//*****
//    function to delete record of file
//*****

void delete_student()
{
    char n[6];
    int flag=0;
    clrscr();
    cout<<"\n\n\n\tDELETE STUDENT...";
    cout<<"\n\nEnter The admission no. of the Student You Want To Delete : ";
    cin>>n;
    fp.open("student.dat",ios::in|ios::out);
    fstream fp2;
    fp2.open("Temp.dat",ios::out);
    fp.seekg(0,ios::beg);
    while(fp.read((char*)&st,sizeof(student)))
    {
        if(strcmpi(st.retadmno(),n)!=0)
            fp2.write((char*)&st,sizeof(student));
        else
            flag=1;
    }

    fp2.close();
    fp.close();
    remove("student.dat");
    rename("Temp.dat","student.dat");
    if(flag==1)
        cout<<"\n\n\tRecord Deleted ..";
}

```

```

        else
            cout<<"\n\nRecord not found";
        getch();
    }

void delete_book()
{
    char n[6];
    clrscr();
    cout<<"\n\n\n\tDELETE BOOK ...";
    cout<<"\n\nEnter The Book no. of the Book You Want To Delete : ";
    cin>>n;
    fp.open("book.dat",ios::in|ios::out);
    fstream fp2;
    fp2.open("Temp.dat",ios::out);
    fp.seekg(0,ios::beg);
    while(fp.read((char*)&bk,sizeof(book)))
    {
        if(strcmpi(bk.retbn(),n)!=0)
        {
            fp2.write((char*)&bk,sizeof(book));
        }
    }

    fp2.close();
    fp.close();
    remove("book.dat");
    rename("Temp.dat","book.dat");
    cout<<"\n\n\tRecord Deleted ..";
    getch();
}

```

```

//*****
//      function to display all students list
//*****

```

```

void display_all()
{
    clrscr();
    fp.open("student.dat",ios::in);
    if(!fp)
    {
        cout<<"ERROR!!! FILE COULD NOT BE OPEN ";
        getch();
        return;
    }
}

```

```

    }

    cout<<"\n\n\t\tSTUDENT LIST\n\n";
    cout<<"===== \n";
    cout<<"\tAdmission No."<<setw(10)<<"Name"<<setw(20)<<"Book Issued\n";
    cout<<"===== \n";

    while(fp.read((char*)&st,sizeof(student)))
    {
        st.report();
    }

    fp.close();
    getch();
}

//*****
//    function to display Books list
//*****

void display_allb()
{
    clrscr();
    fp.open("book.dat",ios::in);
    if(!fp)
    {
        cout<<"ERROR!!! FILE COULD NOT BE OPEN ";
        getch();
        return;
    }

    cout<<"\n\n\t\tBook LIST\n\n";
    cout<<"=====
= \n";
    cout<<"Book Number"<<setw(20)<<"Book Name"<<setw(25)<<"Author\n";
    cout<<"=====
= \n";

    while(fp.read((char*)&bk,sizeof(book)))
    {
        bk.report();
    }
    fp.close();
    getch();
}

```

```

//*****
//      function to issue book
//*****

void book_issue()
{
    char sn[6],bn[6];
    int found=0,flag=0;
    clrscr();
    cout<<"\n\nBOOK ISSUE ...";
    cout<<"\n\n\tEnter The student's admission no.";
    cin>>sn;
    fp.open("student.dat",ios::in|ios::out);
    fp1.open("book.dat",ios::in|ios::out);
    while(fp.read((char*)&st,sizeof(student)) && found==0)
    {
        if(strcmpi(st.retadmno(),sn)==0)
        {
            found=1;
            if(st.rettoken()==0)
            {
                cout<<"\n\n\tEnter the book no. ";
                cin>>bn;
                while(fp1.read((char*)&bk,sizeof(book))&& flag==0)
                {
                    if(strcmpi(bk.retbn(),bn)==0)
                    {
                        bk.show_book();
                        flag=1;
                        st.addtoken();
                        st.getstbn(bk.retbn());
                        int pos=-1*sizeof(st);
                        fp.seekp(pos,ios::cur);
                        fp.write((char*)&st,sizeof(student));
                        cout<<"\n\n\t Book issued successfully\n\nPlease Note:
                                in backside of your book and submit
                                after 15 days
                                ";
                    }
                }
            }
            if(flag==0)
                cout<<"Book no does not exist";
        }
        else
            cout<<"You have not returned the last book ";
    }
}

```

```

    }
    if(found==0)
        cout<<"Student record not exist...";
    getch();
    fp.close();
    fp1.close();
}

//*****
//    function to deposit book
//*****

void book_deposit()
{
    char sn[6],bn[6];
    int found=0,flag=0,day,fine;
    clrscr();
    cout<<"\n\nBOOK DEPOSIT ...";
    cout<<"\n\n\tEnter The student's admission no.";
    cin>>sn;
    fp.open("student.dat",ios::in|ios::out);
    fp1.open("book.dat",ios::in|ios::out);
    while(fp.read((char*)&st,sizeof(student)) && found==0)
    {
        if(strcmpi(st.retadmno(),sn)==0)
        {
            found=1;
            if(st.rettoken()==1)
            {
                while(fp1.read((char*)&bk,sizeof(book))&& flag==0)
                {
                    if(strcmpi(bk.retbn(),st.retstbn())==0)
                    {
                        bk.show_book();
                        flag=1;
                        cout<<"\n\nBook deposited in no. of days";
                        cin>>day;
                        if(day>15)
                        {
                            fine=(day-15)*1;
                            cout<<"\n\nFine has to deposited Rs. "<<fine;
                        }

                        st.resettoken();
                        int pos=-1*sizeof(st);
                        fp.seekp(pos,ios::cur);
                        fp.write((char*)&st,sizeof(student));
                        cout<<"\n\n\t Book deposited successfully";
                    }
                }
            }
        }
    }
}

```

```

        }
        if(flag==0)
            cout<<"Book no does not exist";
        }
        else
            cout<<"No book is issued..please check!!";
    }
}
if(found==0)
    cout<<"Student record not exist...";
    getch();
fp.close();
fp1.close();
}

//*****
//      INTRODUCTION FUNCTION
//*****

void intro()
{
    clrscr();
    gotoxy(35,11);
    cout<<"LIBRARY";
    gotoxy(35,14);
    cout<<"MANAGEMENT";
    gotoxy(35,17);
    cout<<"SYSTEM";
    cout<<"\n\nMADE BY : YOUR NAME";
    cout<<"\n\nSCHOOL : SCHOOL NAME";
    getch();
}

//*****
//      ADMINISTRATOR MENU FUNCTION
//*****

void admin_menu()
{
    clrscr();
    int ch2;
    cout<<"\n\n\n\tADMINISTRATOR MENU";
    cout<<"\n\n\n\t1.CREATE STUDENT RECORD";

```



```

cout<<"\n\n\t2.DISPLAY ALL STUDENTS RECORD";
cout<<"\n\n\t3.DISPLAY SPECIFIC STUDENT RECORD ";
cout<<"\n\n\t4.MODIFY STUDENT RECORD";
cout<<"\n\n\t5.DELETE STUDENT RECORD";
cout<<"\n\n\t6.CREATE BOOK ";
cout<<"\n\n\t7.DISPLAY ALL BOOKS ";
cout<<"\n\n\t8.DISPLAY SPECIFIC BOOK ";
cout<<"\n\n\t9.MODIFY BOOK ";
cout<<"\n\n\t10.DELETE BOOK ";
cout<<"\n\n\t11.BACK TO MAIN MENU";
cout<<"\n\n\tPlease Enter Your Choice (1-11) ";
cin>>ch2;
switch(ch2)
{
    case 1: clrscr();
            write_student();break;
    case 2: display_all();break;
    case 3:
            char num[6];
            clrscr();
            cout<<"\n\n\tPlease Enter The Admission No. ";
            cin>>num;
            display_sps(num);
            break;
    case 4: modify_student();break;
    case 5: delete_student();break;
    case 6: clrscr();
            write_book();break;
    case 7: display_allb();break;
    case 8: {
            char num[6];
            clrscr();
            cout<<"\n\n\tPlease Enter The book No. ";
            cin>>num;
            display_spb(num);
            break;
            }
    case 9: modify_book();break;
    case 10: delete_book();break;
    case 11: return;
    default:cout<<"\a";
}
admin_menu();
}

//*****
// THE MAIN FUNCTION OF PROGRAM

```

```

//*****

void main()
{
    char ch;
    intro();
    do
    {
        clrscr();
        cout<<"\n\n\n\tMAIN MENU";
        cout<<"\n\n\t01. BOOK ISSUE";
        cout<<"\n\n\t02. BOOK DEPOSIT";
        cout<<"\n\n\t03. ADMINISTRATOR MENU";
        cout<<"\n\n\t04. EXIT";
        cout<<"\n\n\tPlease Select Your Option (1-4) ";
        ch=getche();
        switch(ch)
        {
            case '1':clrscr();
                book_issue();
                break;
            case '2':book_deposit();
                break;
            case '3':admin_menu();
                break;
            case '4':exit(0);
            default :cout<<"\a";
        }
    }while(ch!='4');
}

//*****
//                               END OF PROJECT
//*****

```