ਜਗਤ ਗੁਰੂ ਨਾਨਕ ਦੇਵ
ਪੰਜਾਬ ਸਟੇਟ ਓਪਨ ਯੂਨੀਵਰਸਿਟੀ
ਪਟਿਆਲਾ

JAGAT GURU NANAK DEV

PUNJAB STATE OPEN UNIVERSITY, PATIALA

(Established by Act No. 19 of 2019 of the Legislature of State of Punjab)

The Motto of the University

(SEWA)

SKILL ENHANCEMENT **EMPLOYABILITY** **WISDOM** **ACCESSIBILITY**

Bachelor of Computer Applications (BCA)
Course Name: Object Oriented Programming Lab
Course Code: BCA-4-03P

ADDRESS: C/28, THE LOWER MALL, PATIALA-147001
WEBSITE: www.psou.ac.in

JAGAT GURU NANAK DEV
PUNJAB STATE OPEN UNIVERSITY PATIALA
(Established by Act No.19 of 2019 of Legislature of the State of Punjab)

PROGRAMME COORDINATOR :
Dr. Monika Pathak
Assistant Professor, School of Sciences and Emerging Technologies
Jagat Guru Nanak Dev Punjab State Open University, Patiala

PROGRAMME CO-COORDINATOR :
Dr. Gaurav Dhiman
Assistant Professor, School of Sciences and Emerging Technologies
Jagat Guru Nanak Dev Punjab State Open University, Patiala

COURSE COORDINATOR :
Dr. Karan Sukhija (Assistant Professor)
School of Sciences and Emerging Technologies
JGND PSOU, Patiala

JAGAT GURU NANAK DEV
PUNJAB STATE OPEN UNIVERSITY PATIALA
(Established by Act No.19 of 2019 of Legislature of the State of Punjab)

PREFACE

Jagat Guru Nanak Dev Punjab State Open University, Patiala was established in Decembas 2019 by Act 19 of the Legislature of State of Punjab. It is the first and only Open Universit of the State, entrusted with the responsibility of making higher education accessible to all especially to those sections of society who do not have the means, time or opportunity to pursue regular education.

In keeping with the nature of an Open University, this University provides a flexible education system to suit every need. The time given to complete a programme is double the duration of a regular mode programme. Well-designed study material has been prepared in consultation with experts in their respective fields.

The University offers programmes which have been designed to provide relevant, skill-based and employability-enhancing education. The study material provided in this booklet is self instructional, with self-assessment exercises, and recommendations for further readings. The syllabus has been divided in sections, and provided as units for simplification.

The Learner Support Centres/Study Centres are located in the Government and Government aided colleges of Punjab, to enable students to make use of reading facilities, and for curriculum-based counselling and practicals. We, at the University, welcome you to be a part of this institution of knowledge.

Prof. G. S. Batra,
Dean Academic Affairs

Bachelor of Computer Applications (BCA)
Semester IV
BCA-4-03P: Object Oriented Programming Lab

Total Marks: 50
External Marks: 35
Internal Marks: 15
Credits: 2
Pass Percentage: 40%

Course: Object Oriented Programming Lab	
Course Code: BCA-4-03P	
Course Outcomes (COs) After the completion of this course, the students will be able to:	
CO1	Develop the ability to apply OOP fundamentals in creating well-structured and readable code.
CO2	Develop Programs for file handling.
CO3	Develop Programs for Operator Overloading.
CO4	Gain practical experience in implementing OOP concepts such as classes, objects, inheritance, and polymorphism in programming assignments.
CO4	Gain proficiency in using programming languages that support OOP to develop applications and solve real-world problems.

Detailed List of Programs:

Programme No.	Name of Program
P1	Create a class with attributes and methods, and then create objects of that class to demonstrate basic OOP principles.
P2	Create a base class and one or more derived classes to demonstrate inheritance and the use of base class members in derived classes.
P3	Create a base class with a virtual function, override the function in a derived class, and demonstrate runtime polymorphism by calling the function through base class pointers.
P4	Create a class with private data members and public member functions to demonstrate encapsulation and data hiding.
P5	Create a class with a constructor and destructor to demonstrate object

	initialization and cleanup.
P6	Overload arithmetic or comparison operators for a class to demonstrate operator overloading.
P7	Create a class with multiple functions of the same name but different parameters to demonstrate function overloading.
P8	Create an abstract class with one or more pure virtual functions to demonstrate abstract classes and interfaces.
P9	Create a program that allows the user to add, delete, modify, and display student records. Use file handling to store and retrieve student data.

Students/Learners can implement Object Oriented Programming (OOP) concepts using any language like C++ or Java or Python

(Part A)

1) Program to swap two values using pointers and reference variables.
2) Program to find Largest and smallest of four numbers using inline functions.
3) Program to check whether the given number is prime or not using function overloading and also use default arguments.
4) Program to find factorial of a number using function overloading (use both direct and recursive methods to find factorial).
5) Program to accept cricket players name, total runs, and total matches and print these details with batting average (Use array of objects).
6) Program in which Create a class to hold information of a husband and another for the wife. Using friend function find the total salary of the family.
7) Program to demonstrate static members.
8) Program to create a database for a bank account contains Name, Account no, Account type, Balance, Including the following a) Constructors b) destructors c) default constructors d) Input and output functions.

(Part B)

9) Program to increment the given date using ++ operator (unary operator).
10) Program to overload the following operators:
    a) Binary operator '+' to concatenate 2 strings
    b) compare to strings using = = operator
11) Program in which Create a base class for a stack and implement push and pop operation. Include a derived class to check for stack criteria such as a) stack empty b) stack full c) stack overflow d) stack underflow.
12) Program to illustrate hybrid inheritance.
13) Program to sort n names using pointer sort.
14) Program to demonstrate Virtual function.
15) Program to create a database using concepts of files for a student including the fields:

Student-name, Student's Register No, Student's Attendance (overall % of attendance) and enter data for n students and output the same in proper format.

16) Program to accessing a particular record in an employee file.

Practice Exercise:

- Write a C++ Program to display Names, Roll No., and grades of 3 students who have appeared in the examination. Declare the class of name, Roll No. and grade. Create an array of class objects. Read and display the contents of the array.
- Write a C++ program to declare Struct. Initialize and display contents of member variables.
- Write a C++ program to declare a class. Declare pointer to class. Initialize and display the contents of the class member.
- Given that an EMPLOYEE class contains following members: data members: Employee number, Employee name, Basic, DA, IT, Net Salary and print data members.
- Write a C++ program to read the data of N employee and compute Net salary of each employee (DA=52% of Basic and Income Tax (IT) =30% of the gross salary).
- Write a C++ to illustrate the concepts of console I/O operations.
- Write a C++ program to use scope resolution operator. Display the various values of the same variables declared at different scope levels.
- Write a C++ program to allocate memory using new operator.
- Write a C++ program to create multilevel inheritance. (Hint: Classes A1, A2, A3)
- Write a C++ program to create an array of pointers. Invoke functions using array objects.
- Write a C++ program to use pointer for both base and derived classes and call the member function. Use Virtual keyword.

Solutions:

```
/* 1. Program to  swap two values using pointers and
     reference variables */

#include<iostream.h>
#include<conio.h>
void swappoint(int *,int *);
void swapref(int &,int &);

void main()
{
int a,b;
clrscr();
cout<<"Enter two numbers"<<endl;
cin>>a>>b;
cout<<"\nValues of A and B before calling function"<<endl;
cout<<"A="<<a<<" B="<<b;
swappoint(&a,&b);
cout<<"\nValues of A and B after calling swappoint()"<<endl;
cout<<"A="<<a<<" B="<<b;
swapref(a,b);
cout<<"\nValues of A and B after calling swapref()"<<endl;
cout<<"A="<<a<<" B="<<b;
getch();
}

void swappoint(int *l,int *m)
{
int k;
k=*l;
*l=*m;
*m=k;
}

void swapref(int &l,int &m)
{
int k;
```

```
k=l;
l=m;
m=k;
}




/* 2.Program to find Largest and smallest of four numbers
     Using inline functions. */
#include<iostream.h>
#include<conio.h>
inline int largest(int x, int y)
{
return(x>y?x:y);
}
inline int smallest(int x, int y)
{
return(x<y?x:y);
}

void main()
{
clrscr();
int a,b,c,d;
cout<<"Enter any four numbers:";
cin>>a>>b>>c>>d;
int lar=largest(largest(a,b),largest(c,d));
cout<<"\n Largest of the given four numbers is:"<<lar;
int small=smallest(smallest(a,b),smallest(c,d));
cout<<"\n Smallest of the given four numbers is:"<<small;
getch();
}

/* 3. Program to check whether the given number is prime or
      not using function overloading and also use default
      arguments.*/
```

```
#include<iostream.h>
#include<conio.h>
#include<stdlib.h>
void prime(), prime(int,int=1);

void main()
{
  int choice, n;

  while(1)
  {
     clrscr();
     cout<<"\n1.Prime number checking without parameters";
     cout<<"\n2.Prime number ckecking with parameters";
     cout<<"\n3.Exit";
     cout<<endl<<"\nEnter your choice:";
     cin>>choice;
     switch(choice)
     {
       case 1:prime();
           getch();
           break;
       case 2:cout<<"Enter any number:";
           cin>>n;
           prime(n);
           getch();
           break;
       case 3:exit(0);
           break;
       default: cout<<"\nInvalid Choice";
          getch();
     }
   }
 }

 void prime()
```

```
{
   int n, flag=1;
   cout<<"Enter any number:";
   cin>>n;
   for(int i=2;i<n/2;i++)
      if(n%i==0)
      {
      flag=0;
      break;
      }
      if(flag==1)
     cout<<endl<<"The Given Number "<<n<<" is a prime number";
      else
     cout<<endl<<"The Given Number "<<n<<" is not a prime
                  number";
}

void prime(int n,int flag)
{
    for(int i=2;i<n/2;i++)
      if(n%i==0)
      {
      flag=0;
      break;
      }
    if(flag==1)
     cout<<endl<<"The Given Number "<<n<<" is a prime number";
      else
     cout<<endl<<"The Given Number "<<n<<" is not a prime
                  number";
}
```

```
/* 4.Program to find factorial of a number using
     function overloading (use both direct and recursive
     methods to find factorial) */

#include<iostream.h>
#include<conio.h>
#include<stdlib.h>
void fact();
long int fact(int);

void main()
{
  int choice, n;
  for(;;)
  {
     clrscr();
     cout<<"1.Factorial of a number without parameters"<<endl;
     cout<<"2.Factorial of a number with parameters"<<endl;
     cout<<"3.Exit"<<endl;
     cout<<endl<<"Enter your choice:";
     cin>>choice;
     switch(choice)
     {
       case 1:fact();
           getch();
           break;
       case 2:cout<<"Enter the number:";
           cin>>n;
           cout<<"\n The factorial of "<<n<<" is "<<fact(n);
           getch();
           break;
       case 3:exit(0);
       default:cout<<"\n Invalid choice";
            getch();
     }
   }
```

```
 }

 void fact()
 {
   int n,i;
   cout<<endl<<"Enter the number:";
   cin>>n;
   long int fact=1;
   for(i=1;i<=n;i++)
   fact=fact*i;
   cout<<endl<<"The Factorial of "<<n<<" is "<<fact;
 }



long int fact( int n)
 {

   if(n==0)
   return 1;
   else
   return n*fact(n-1);
 }

/* 5.Program to accept cricket players name, total
     runs, and total matches and print these details with
     batting average(Use array of objects). */

#include<stdio.h>
#include<conio.h>
#include<iostream.h>
#include<iomanip.h>

class cricket
{
char name[30];
float truns;
```

```
float tmat;
float avg;
public:
void get_data();
void put_data();
};

void cricket::get_data()
{
cout<<"Enter the player name:"<<endl;
cin>>name;
cout<<"Enter total runs of the player "<<name<<":"<<endl;
cin>>truns;
cout<<"Enter total matches played by "<<name<<":"<<endl;
cin>>tmat;
avg=truns/tmat;
}

void cricket::put_data()
{
cout.setf(ios::left,ios::adjustfield);
cout<<setw(10)<<name<<"\t"<<setw(10)<<truns<<"\t"
 <<setw(10)<<tmat<<"\t"<<setw(10)<<setprecision(2)<<avg<<endl;
}

void main()
{
clrscr();
int n;
cricket e[' '];
cout<<"Enter number of players:";
cin>>n;
for(int i=1;i<=n;i++)
{
cout<<"Enter the details of player "<<i<<endl;
e[i].get_data();
}
```

```
clrscr();
cout<<"-------------------------------------------------------
       -------- "<<endl;
cout<<"Player-Name\tTotal-Runs\tTotal-Matches\tBatting-
       Average\n";
cout<<"-------------------------------------------------------
       --------"<<endl;
for(i=1;i<=n;i++)
{
e[i].put_data();
}
cout<<"-------------------------------------------------------
       --------";
getch();
}

/* 6. Program in which Create a class to hold
     information of a husband and another for the wife. Using
     friend function find the total salary of the family. */

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

class wife;

class husband
{
char name[10];
float sal;

public:

void read()
{
cout<<"\nEnter Name and Salary of Husband:" ;
cin>>name>>sal;
}
```

```
void print()
{
cout<<"\nName:"<<name<<"\nSalary:"<<sal<<endl;
}
friend float total(husband,wife);
};

class wife
{
char name[10];
float sal;

public:

void read()
{
cout<<"\nEnter Name and Salary of Wife:";
cin>>name>>sal;
 }

void print()
{
cout<<"\nName:"<<name<<"\nSalary:"<<sal<<endl;
}
friend float total(husband,wife);
};

float total(husband h,wife w)
{
return(h.sal+w.sal);
}

void main()
{
husband h;
wife w;
```

```
clrscr();
h.read();
w.read();
clrscr();
cout<<"\nHusband Details:";
h.print();
cout<<"\nWife Details:";
w.print();
cout<<"\nTotal Monthly Income of the Family:"<<total(h,w);
getch();
}
```

```
/* 7. Program to demonstrate static members */


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

class count
{
private :
static int number;
int idno;
public:

void assignidno()
{
idno=++number;
}
```

```
void display()
{
cout<<"object number :"<<idno<<endl;
}

static void displaycount()
{
cout<<"Number of objects created so far :"<<number<<endl;
}
};

int count::number;

void main()
{
count obj1;
clrscr();
cout<<"\n Program to demonstrate static members \n\n";
obj1.assignidno();
count::displaycount();
count obj2,obj3;
obj2.assignidno();
obj3.assignidno();
count::displaycount();
obj1.display();
obj2.display();
obj3.display();
getch();
}
```

```
/* 8. Program to create a database for a bank account contains
Name, Account no, Account type, Balance, Including the
following a) Constructors  b) destructors  c) default
constructors d) Input and output functions. */

   #include<iostream.h>
   #include<conio.h>
   #include<string.h>
   #include<iomanip.h>
   class bank
   {
     private:
       char name[20];
       int ano;
       char atype[2];
       float balance;
     public:
       bank(){}
       bank(char na[], int no,char type[],float bal)
       {
         strcpy(name, na);
         ano=no;
         strcpy(atype,type);
         balance=bal;
       }
       void readdata()
       {
         cout<<endl<<"Name :";
         cin>>name;
         cout<<endl<<"Account Number :";
         cin>>ano;
         cout<<endl<<"Account type :";
         cin>>atype;
         cout<<endl<<"Balance :";
         cin>>balance;
       }
```

```
    void writedata()
    {

cout<<endl<<setw(15)<<name<<setw(7)<<ano<<setw(9)<<atype
    <<setw(15)<<balance;
    }
    ~bank()
    {
    }
  };
  void main()
  {
    int n;
    bank cust[' '],cust1("Ashok",1220,"SB",10000.00);
    clrscr();
    cout<<"Enter number of customers:";
    cin>>n;
    for(int i=1;i<=n;i++)
    cust[i].readdata();
    clrscr();
    cout.setf(ios::left,ios::adjustfield);
    cout<<"\n \n Database for Bank Account \n \n ";
    cout<<"Name          Number Type     Balance \n";
    cust1.writedata();
    for(i=1;i<=n;i++)
    cust[i].writedata();
    getch();
  }
```

PART-B

```
/*9 Program to increment the given date using ++ operator
(unary operator)*/


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

class udate
{
int day,month,year;
public:

void read()
{
cin>>day>>month>>year;
}

void write()
{
cout<<day<<"/"<<month<<"/"<<year;
}
void operator ++();
};

void udate :: operator ++()
{
day++;
if(day>31)
{
   if(month==12)
   {
    day=1;month=1;year++;
   }
    else
   {
```

```
    day=1;
     month++;
   }
}
if((month==4||month==6||month==9||month==11)&&(day>30))
{
day=1;month++;
}

if(month==2)
{
   if(year%4==0 && day>29)
     {
     day=1;
     month++;
     }
    else
     if (year%4!=0 && day >28)
     {
      day=1;
      month++;
      }
}
}

void main()
{
udate d1;
clrscr();
cout<<"Enter the date:";
d1.read();
cout<<"\n The given date is:";
d1.write();
++d1;
cout<<endl<<"\n The incremented date is : ";
d1.write();
getch();
```

```
}


/* 10. program to overload  the following operators

        a) Binary operator '+' to concatenate 2 strings
        b)  compare to strings  using = = operator  */

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

class string
{
private:
       char str[80];
public:

       void getstring()
       {
       cin.getline(str,80);
       }


      void display()
       {
       cout<<str;
       }
       string operator +(string);
       int operator ==(string);
};

string string::operator +(string ss)
{
 string temp;
 strcpy(temp.str,str);
 strcat(temp.str,ss.str);
 return temp;
}
```

```
int string::operator ==(string ss)
{
return(strcmp(str,ss.str));
}

void main()
{
string s1,s2,s3;
clrscr();
cout<<"Enter the first string :";
s1.getstring();
cout<<endl<<"Enter the second string :";
s2.getstring();
s3=s1+s2;
clrscr();
cout<<endl<<"\n The first string is:";
s1.display();
cout<<endl<<"\n The second string is :";
s2.display();
cout<<endl<<"\n The concatenation of two string :";
s3.display();
if(s1==s2)
cout<<"\n The given two strings  are not equal";
else
cout<<"\n The given two strings are equal";
getch();
}


/* 11.Program in which Create a base class for a stack and
implement push and pop operation.  Include a derived class to
check for stack criteria such as a) stack empty b) stack full
c) stack overflow d) stack underflow. */

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

```
#include<stdio.h>
#define max 5

class stack
{
protected:
      int st[max];
      int top;
public:
       stack()
       {
        top = -1;
       }

       void push()
       {
       int item;
       cout<<"enter the item to push:";
       cin>>item;
       top++;
       st[top]=item;
       }

       int pop()
      {
       int item;
       item=st[top];
       top--;
       return(item);
      }

      void display()
      {
      cout<<"\n the stack contains \n"<<endl;
      for(int i=top;i>=0;i--)
      cout<<st[i]<<endl;
      }
```

```
     };

class stack_cond:public stack
{
public:

void push()
{
if(top==max-1)
cout<<"\n stack over flow"<<endl;
else
{
stack::push();
if(top == max-1)
cout<<endl<<" after push stack is full"<<endl;
}
}

int pop()
{
int item;
if(top == -1)
{
cout<<endl<<"stack underflow"<<endl;
return(0);
}
else
{
item=stack::pop();
if(top == -1)
cout<<endl<<"after pop stack is empty"<<endl;
return(item);
}
}

void display()
{
```

```
if(top == -1)
cout<<"stack is empty"<<endl;
else
stack::display();
}
};

void main()
{
stack_cond s;
int choice;
cout<<"\n program for stack using inhertance \n\n";
do
{
clrscr();
cout<<"1. push \n";
cout<<"2. pop \n";
cout<<"3. display \n";
cout<<"4. exit \n";
cout<<" \n enter your choice:";
cin>>choice;
switch(choice)
{
case 1: s.push();
     s.display();
     break;
case 2: s.display();
     choice=s.pop();
     if(choice!=0)
     cout<<endl<<"the items "<<choice<<" is poped from the top
of stack"<<endl;
     break;
case 3: s.display();
     break;
case 4: exit(0);
}
cout<<"press any key to continue....."<<endl;
```

```
getch();
}
while(choice!=4);
}



/* 12. Program to illustrate hybrid inheritance */

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

   class student
     {
          int regno;
          char name[20];
       public:
        void input()
        {
        cout<<"Enter Register Number & Name:\n";
        cin>>regno>>name;
        }
        void output()
        {
        cout<<"\nName:"<<name;
        cout<<"\nRegno:"<<regno;
        }
      };

      class test: public student
       {
          protected:
          int m1,m2,m3;
          public:
          void input()
          {
          cout<<"\nEnter 3 subject marks\n";
```

```
        cin>>m1>>m2>>m3;
        }
        void output()
        {
        cout<<"\nSubject1 Marks:"<<m1;
        cout<<"\nSubject2 Marks:"<<m2;
        cout<<"\nSubject3 Marks:"<<m3;
        }
    };

     class sports
      {
        protected:
        int wt;
        public:
        void input()
        {
        cout<<"Enter Sports Weightage\n";
        cin>>wt;
        }

      void output()
        {
        cout<<"\nSports Weightage:"<<wt;
        }
    };

     class result: public test,sports
        {
        int total;
        public:

        void caltotal()
        {
        student::input();
        test::input();
        sports::input();
```

```
            total=m1+m2+m3+wt;
            clrscr();
            student::output();
            test::output();
            sports::output();
            cout<<"\nTotal marks:"<<total;
            }
     };

          void main()
            {
            clrscr();
            result r;
            r.caltotal();
            getch();
            }



/* 13. Program to sort n names using pointer sort */

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

class student
{
private:
char name[15];
public:

void getdata()
{
cin>>name;
}

void putdata()
```

```
{
cout<<name<<endl;
}

char* getname()
{
return name;
}
};

void main()
{
int n;
void sort(student**,int);
student *stu[10];
clrscr();
cout<<"Enter the total number of students:";
cin>>n;

for(int i=0;i<n;i++)
{
cout<<"\n Enter the name of the student:"<<i<<endl;
stu[i]=new student;
stu[i]->getdata();
}
clrscr();
cout<<"\n Before sorting : \n";
cout<<"\n The names are :\n";
for(i=0;i<n;i++)
stu[i]->putdata();
sort(stu, n);
cout<<"\n After sorting:\n";
cout<<"\n The names are:\n";
for(i=0;i<n;i++)
stu[i]->putdata();
getch();
}
```

```
void sort(student** s,int n)
{
void swap(student**, student**);
for(int i=0;i<n;i++)
for(int j=i+1;j<n;j++)
swap(s+i, s+j);
}

void swap(student** s1, student** s2)
{
if(strcmp((*s1)->getname(),(*s2)->getname())>0)
{
student *temp = *s1;
*s1 = *s2;
*s2 = temp;
}
}


/* 14. Program to demonstrate Virtual function.*/

#include<iostream.h>
#include<stdio.h>
#include<conio.h>
#include<string.h>

class media
{
  protected:
  char title[20];
  double price;
  public:
  media(char *s,double p)
  {
  strcpy(title,s);
  price=p;
```

```
  }
  virtual void display()=0;
};

class book: public media
{
   int noofpages;
   public:
   book(char *s,double a,int p):media(s,a)
   {
   noofpages=p;
   }
   void display()
   {
   cout<<"\nTitle:"<<title;
   cout<<"\nPages:"<<noofpages;
   cout<<"\nPrice:"<<price;
   }
};

class tape: public media
{
   int playtime;
   public:
   tape(char *s,double a,int pt): media(s,a)
   {
   playtime=pt;
   }
   void display()
   {
   cout<<"\nTitle:"<<title;
   cout<<"\nPlaytime:"<<playtime<<" Minutes";
   cout<<"\nPrice:"<<price;
   }
};

void main()
```

```
{
     clrscr();
     char t[100];
     double pr;
     int pt,nop;
     media *m;
     cout<<"Enter Title,NoofPages & price of a Book:\n";
     gets(t);
     cin>>nop>>pr;
     book b(t,pr,nop);
     cout<<"Enter Title,Price & Playtime(in minutes) of a
            Tape:\n";
     gets(t);
     cin>>pr>>pt;
     tape T(t,pr,pt);
     m=&b;
     clrscr();
     cout<<"\nBook Information";
     m->display();
     m=&T;
     cout<<"\n\nTape Information";
     m->display();
     getch();
  }


/* 15. Program to create a database using concepts of files
for a student including the fields: Student-name, Student's
Register No, Student's Attendance (overall % of attendance)
and enter data for n students and output the same in proper
format*/


#include<iostream.h>
#include<conio.h>
#include<fstream.h>
#include<iomanip.h>
#include<stdio.h>
```

```
class student
{
public:
int regno;
char name[15];
float attendance;
};



void line(int n)
{
for(int i=0;i<=n;i++)
cout<<"_";
}

void main()
{
student stu;
int n;
ofstream inputfile;
inputfile.open("stud.dat",ios::out/ios::binary);
clrscr();
cout<<"Enter number of students:";
cin>>n;
cout<<"Enter the student details for "<<n<<" students"<<endl;
for(int i=1;i<=n;i++)
{
clrscr();
fflush(stdin);
cout<<"Record no:"<<i<<endl;
cout<<endl<<"Regno:";
cin>>stu.regno;
cout<<endl<<"Name:";
cin>>stu.name;
cout<<endl<<"Attendance:";
cin>>stu.attendance;
```

```
inputfile.write((char*)&stu,sizeof(stu));
}
inputfile.close();
ifstream outputfile;
outputfile.open("stud.dat",ios::in/ios::binary);
clrscr();
int row = 5;
gotoxy(7, row++);
line(50);
gotoxy(7,row++);
cout<<setw(40)<<"Student Attendance Report"<<endl;
gotoxy(7,row++);
cout<<setw(9)<<"Reg.No"<<setw(15)<<"Name"<<setw(15)<<"Attendan
ce";
gotoxy(7,row++);
line(45);
fflush(stdout);
while(outputfile.read((char*)&stu,sizeof(stu)))
{
gotoxy(7,row++);
cout<<setw(5)<<stu.regno;
cout<<setw(20)<<stu.name;
cout<<setw(10)<<stu.attendance;
}
outputfile.close();
gotoxy(7,row++);
line(45);
getch();
}


/* 16.Program to accessing a particular record in an employee
      file */

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

```
#include<iomanip.h>
#include<stdio.h>
class employee
{
private:
       int empno;
       char name[15];
       long int salary;
public:
       void getdata()
       {
       fflush(stdin);
       cout<<"Enter employee number,name and salary:\n";
       cin>>empno>>name>>salary;
       }
       void putdata()
       {
         cout<<setw(5)<<empno<<setw(10)<<name<<setw(10)
             <<salary<<endl;
       }
};


void main()
{
fstream f1;
employee emp;
char ch;
int n;
clrscr();
f1.open("emp.dat",ios::out);

do
{
emp.getdata();
f1.write((char*)&emp,sizeof(emp));
cout<<"\n Do you want to continue(y/n) ?";
```

```
cin>>ch;
}
while(ch=='y'||ch=='Y');

f1.close();
f1.open("emp.dat",ios::in);
clrscr();
cout<<"\n Employee details \n\n";
cout<<setw(5)<<"Empno"<<setw(10)<<"Name"<<setw(10)<<"Salary"<<
      endl;
while(f1.read((char*)&emp,sizeof(emp)))
emp.putdata();
cout<<"\n Enter the record number you want to view:";
cin>>n;
f1.close();
f1.open("emp.dat",ios::in);
int p = (n-1)*sizeof(emp);
f1.seekg(p);
f1.read((char*)&emp,sizeof(emp));
emp.putdata();
f1.close();
getch();
}
```