COMPUTER SCIENCE PROJECT FILE

NAME: ARJUN SOOTA

CLASS: XII-A

BOARD ROLL NO: 9147425

CERTIFICATE

This is to certify that ARJUN SOOTA (9147425) of class XII-A has completed the computer science project entitled "Payroll Systems" himself and under my guidance and to my complete satisfaction.The process of the project had been continuously reported and he has been in my acknowledgement constantly.

Ms. Niti Arora

(P.G.T. Computer Science)

ACKNOWLEDGEMENT

It gives me great pleasure to express my gratitude towards my computer science teacher MS. NITI ARORA for her guidance and support throughout my project. Without her help, support and motivation, project could not be completed successfully.

INDEX

OVERVIEW OF C++

C++ is an "object oriented" programming language created by Bjarne Stroustrup at Bell Labs and released in 1985. It implements "data abstraction" using a concept called "classes", along with other features to allow object-oriented programming. Parts of the C++ program are easily reusable and extensible; existing code is easily modifiable without actually having to change the code. C++ adds a concept called "operator overloading" not seen in the earlier OOP languages and it makes the creation of libraries much cleaner.

It has imperative, object-oriented and generic programming features, while also providing facilities for low-level memory manipulation. Additionally, some of the features of C++ allow low-level access to memory but also contain high level features.

C++ could be considered a superset of C. C programs will run in C++ compilers. C uses structured programming concepts and techniques while C++ uses object oriented programming and classes which focus on data. C++ is a general purpose programming language with a bias towards systems programming that

- is better than C
- supports object-oriented programming

Some of the important object oriented features are namely:

Objects:

Object is the basic unit of object-oriented programming. Objects are identified by its unique name. An object represents a particular instance of a class. There can be more than one instance of an object. Each instance of an object can hold its own relevant data.

Classes:

Classes are data types based on which objects are created. Objects with similar properties and methods are grouped together to form a Class. Thus a Class represents a set of individual objects. Characteristics of an object are represented in a class as Properties. The actions that can be performed by objects become functions of the class and are referred to as Methods.

Inheritance:

Inheritance is the process of forming a new class from an existing class or *base class*. The base class is also known as *parent class* or *super class*, the new class that is formed is called *derived class*. Derived class is also known as a *child class* or *sub class.* Inheritance helps in reducing the overall code size of the program, which is an important concept in object-oriented programming.

Data Abstraction:

Data Abstraction increases the power of programming language by creating user defined data types. Data Abstraction also represents the needed information in the program without presenting the details.

Data Encapsulation:

Data Encapsulation combines data and functions into a single unit called class. When using Data Encapsulation, data is not accessed directly; it is only accessible through the functions present inside the class. Data Encapsulation enables the important concept of data hiding possible.

Polymorphism:

Polymorphism allows routines to use variables of different types at different times. An operator or function can be given different meanings or functions. Polymorphism refers to a single function or multi-functioning operator performing in different ways.

Overloading:

Overloading is one type of Polymorphism. It allows an object to have different meanings, depending on its context. When an exiting operator or function begins to operate on new data type, or class, it is understood to be overloaded.

Reusability:

This term refers to the ability for multiple programmers to use the same written and debugged existing class of data. This is a time saving device and adds code efficiency to the language. Additionally, the programmer can incorporate new features to the existing class, further developing the application and allowing users to achieve increased performance. This time saving feature optimizes code, helps in gaining secured applications and facilitates easier maintenance on the application.

SYNOPSIS

Project Category: Human Resource

Target Group: Can be used by any company for Salary Management of their employees

Project Members: Arjun Soota
Mehul Bhagi

DESCRIPTION:

1. This software can used to process the salary for the employees on monthly or yearly basis.
2. First of all master code of each employee is to be created with basic data like name, father's name, designation etc.
3. A binary master file is created which maintains data like :
    - Basic salary and Net salary
    - HRA rate
    - DA rate
    - PF rate
    - Tax
4. Basis the above input system will then calculate the gross and net salary for an employee using following formula:

    Gross salary = Basic +HRA +DA

    Net salary = Gross – PF–Tax
5. The system also provides functions to input the attendance records of employees.
6. Salary slip is then generated gathering data from both master file and attendance record file.

SYSTEM REQUIREMENTS

The Project requires the following:

- Hardware
    - Computer System (Pent-II or above with standard configuration)
    - Keyboard
    - Mouse
- Software
    - DOS – 6.22 /Windows OS (any version)
    - Turbo C++

HEADER FILES

In computer programming, particularly in the C and C++ programming languages, a header file or include file is a file, usually in the form of source code that is automatically included in another source file by the compiler. Typically, header files are included via compiler directives at the beginning (or *head*) of the other source file.

A header file commonly contains forward declarations of subroutines, variables, and other identifiers. Identifiers that need to be declared in more than one source file can be placed in one header file, which is then included whenever its contents are required.

In the C and C++ programming languages, standard library functions are traditionally declared in header files

- include<graphics.h> : cleardevice(),setbkcolor(),line(),bar(),rectangle()
- #include <fstream.h> : read(), write(), open(), close(), seekg(), seekp(), tellg( )
- #include <string.h> : strcpy(), strlen(), strcmpi()
- #include<iomanip.h> : setw(), setprecision()
- #include <stdio.h> : gets()
- #include <conio.h> : getch(), clrscr(),gotoxy( )
- #include <dos.h> : delay()
- #include <process.h> : exit()
- #include <ctype.h> : toupper()

PROGRAM FILES

1. payroll.cpp : The main program which is to be executed for creating and maintenance of payroll system.

DATA FILES

1. emp.dat: binary master file for storing all the data of employees
2. att.dat: binary file for storing the attendance records of employees

GLOBAL CLASSES and OBJECTS (payroll.cpp)

```
void  click()
{
          callmouse();
          do
          {
               mouseposi(x,y,cl);
            }while(cl!=1);
  }
class employee
{
       int empno;
       char Name[30],Faname[20],Qualification[30],Mobile[12];
       int exp;//experience
       public:
       float salary;
       char Department[20],Designation[20];
       float Basic, DA, Hra, PF,Gross,Net,anngross,Tax;
       /*gross=basic+hra+da and basic is standard
        rate of pay before additional payment such as allowances .
        net=gross-(pf+tax) and then calculate tax */
       void getdata();
       void dispdata();
       void display2();
       void report();
       void Salary();
       int Return_empno()
       {return empno;
       }
       char *Return_name()
       {
             return Name;
       }
       char *Return_Name()
       {return Name;
       }
            };
employee ob1;//global object of class employee is declared and is named ob1

class attendance
{
   public:
   int Empno,casleave,earnleave,earnleaveavail,casleaveavail,LWP;
   void Agetdata();
   void Adisplaydata();
  };

attendance att;//object of class attendance created
```

SOURCE CODE

```
#include<fstream.h>
#include<conio.h>
#include<dos.h>
#include<iomanip.h>
#include<stdio.h>
#include<string.h>
#include<process.h>
#include<graphics.h>

char agn,agn1,agn2,agn3,agn4,rep,reply;

//*****CODING FOR MOUSE CONTROLS**************

union REGS in,out;
int x,y,cl;

      int callmouse()
      {
            in.x.ax=1;
            int86(0x33,&in,&out);
            return 1;
      }
      void mouseposi(int &xpos,int &ypos,int &click)
      {
            in.x.ax=3;
            int86(0x33,&in,&out);
            click=out.x.bx;
            xpos=out.x.cx;
            ypos=out.x.dx;
       }
void  click()

{
   callmouse();
   do
    {
      mouseposi(x,y,cl);
       }while(cl!=1);
     }//the while loop will run until the user presses the right click i.e. Cl=1


class employee
{
       int empno;
       char Name[30],Faname[20],Qualification[30],Mobile[12];
       int exp;//experience
       public:
       float salary;
```

```
        char Department[20],Designation[20];
        float Basic, DA, Hra, PF,Gross,Net,anngross,Tax;
        /*gross=basic+hra+da and basic is standard
         rate of pay before additional payment such as allowances .
         net=gross-(pf+tax) and then calculate tax */
        void getdata();
        void dispdata();
        void display2();
        void report();
        void Salary();
        int Return_empno()
        {return empno;
        }
        char *Return_name()
        {
               return Name;
        }
              };//class ended

employee ob1; //object of class employee is declared and is named ob1

class attendance
{   public:
    int Empno,casleave,earnleave,earnleaveavail,casleaveavail,LWP;
    void Agetdata();
    void Adisplaydata();
  };

attendance att; //object of class attendance created

void employee::getdata()
{
  cleardevice();
  gotoxy(25,1);
  cout<<"\n\nEnter Employee ID : "  ;
  cin>>empno;
  cout<<"\nEnter Employee Name : ";
  gets(Name);
  cout<<"\nEnter Employee's Father's Name:  ";
  gets(Faname);
  cout<<"\nEnter the Qualification of the employee:  ";
  gets(Qualification);
  cout<<"\nEnter mobile number of the employee:  ";
  gets(Mobile);
  cout<<"\nEnter the experience of the employee in years:  ";
  cin>>exp;
  cout<<"\nEnter the Department of the employee : ";
  gets(Department);
  cout<<"\nEnter the designation of the employee : ";
  gets(Designation);
  cout<<"\nEnter the basic salary of the employee : ";
  cin>>Basic;
```

```
  cout<<"\nEnter the hra rate : ";
  cin>>Hra;
  cout<<"\nEnter the DA rate  :  ";
  cin>>DA;
  cout<<"\nEnter the PF rate  : ";
  cin>>PF;

   //CALCULATIONs .....
         Hra=(Hra*Basic)/100;
         DA=(DA*Basic)/100;
         PF=(PF*Basic)/100;
         Gross=Basic+Hra+DA;
         anngross=Gross*12.0;
       if (anngross<100000.0)
          Tax=0.0;
         else if(anngross<=250000.0)
          Tax=(0.15)*Gross;
         else if(anngross<=500000.0)
           Tax=(0.20)*Gross;
         else
           Tax=(0.25)*Gross;
      Net=Gross-PF-(Tax/12.0);
   }
void employee::dispdata()
{
  cout<<"\n\nEmoployee id is :"<<empno;
  cout<<"\nEmployee name is :"<<Name;
  cout<<"\nemployee's father's name:"<<Faname;
  cout<<"\nthe qualification of the eployee is :"<<Qualification;
  cout<<"\nthe moblie number of employee is :"<<Mobile;
  cout<<"\nthe experince of employee is :"<<exp<<"years";
  cout<<"\nthe designation of the employee is : "<<Designation;
  cout<<"\nthe department of the employee is : "<<Department;
  cout<<"\nthe basic salary of the employee is : "<<Basic;
  cout<<"\nThe HRA of the Employee is : "<<Hra;
  cout<<"\nThe PF of the employee is : "<<PF;
  cout<<"\nThe DA of the employee is : "<<DA;
  cout<<"\nThe NET Annual income of the employee is : "<<Net;
  cout<<"\nThe TAX deducted is : "<<Tax;
 }

void loading()
{  cleardevice();
   gotoxy(37.25,14);
      cout<<"Loading....";
   /* loop through the fill patterns */
   for (int i,j=0; j<=100,i<170;j+=10,i+=15)
   {
     setfillstyle(SOLID_FILL,15);
      /* draw the bar */
       bar(230+i,200,250+i,215);
       gotoxy(39.25,17);
```

```
      cout<<j<<"%";
      delay(100);

  }
}

//*****************CODING FOR BMP************************

void colo(int);
int bmp(char *,int,int,int,int);
int CANVASSX=50,CANVASSY=40,CANVASEX=640,CANVASEY=480;
int colo1,colo2;

void colo(int no)
{
  int num[8]={0,0,0,0,0,0,0,0},i=0;
  while(no!=0)
  {
    num[i]=no%2;
    no=no/2;
    i++;
  }
  colo1=8*num[7]+4*num[6]+2*num[5]+num[4];
  colo2=8*num[3]+4*num[2]+2*num[1]+num[0];
}

void canvasresize(int px=0,int py=0)
{
  if(CANVASEX>600)
  CANVASEX=600;
  if(CANVASEY>450)
  CANVASEY=450;
  if(px!=0||py!=0)
  {
    setfillstyle(1,15);
    if(px<CANVASEX)
    bar(px,CANVASSY,CANVASEX,py);
    if(py<CANVASEY)
    bar(CANVASSX,py,CANVASEX,CANVASEY);
  }
  setfillstyle(1,7);
  if(CANVASEX!=600)
  bar(CANVASEX+1,CANVASSY+1,599,449);
  if(CANVASEY!=450)
  bar(CANVASSX+1,CANVASEY+1,599,449);
  setcolor(0);
  rectangle(CANVASSX,CANVASSY,CANVASEX,CANVASEY);
}

void initiatebmp()
{
  struct palettetype pal;
```

```
  getpalette(&pal);
  setrgbpalette(pal.colors[1],96,0,0);        // MAROON
  setrgbpalette(pal.colors[2],0,96,0);        // DARK GREEN
  setrgbpalette(pal.colors[3],96,96,0);       // DARK YELLOW
  setrgbpalette(pal.colors[4],0,0,96);        // DARK BLUE
  setrgbpalette(pal.colors[5],96,0,96);       // MAGENTA
  setrgbpalette(pal.colors[6],0,96,96);       // CYAN
  setrgbpalette(pal.colors[7],48,48,48);      // LIGHT GRAY
  setrgbpalette(pal.colors[8],96,96,96);      // DARK GRAY
  setrgbpalette(pal.colors[9],63,0,0);        // RED
  setrgbpalette(pal.colors[10],0,63,0);       // GREEN
  setrgbpalette(pal.colors[11],63,63,0);      // YELLOW
  setrgbpalette(pal.colors[12],0,0,63);       // BLUE
  setrgbpalette(pal.colors[13],63,0,63);      // PINK
  setrgbpalette(pal.colors[14],0,63,63);      // LIGHT CYAN
}

int bmp(char *ptr,int shiftx,int shifty,int use=0,int type=0)
{
  char ch;
  long i,filesize=0,blankspace,Xlimit=0,Ylimit=0;
  ifstream fi;
  fi.open(ptr,ios::binary);
  fi.get(ch);
  while(fi)
  {
    filesize++;
    fi.get(ch);
  }
  fi.close();
  fi.open(ptr,ios::binary);
  fi.get(ch);
  if(ch!='B')
   return 0;
  fi.get(ch);
  if(ch!='M')
   return 0;
  fi.seekg(18);
  fi.get(ch);
  Xlimit=int(ch);
  if(Xlimit<0)
   Xlimit=256+Xlimit;
  fi.get(ch);
  if(int(ch)<0)
   Xlimit=256*(256+int(ch))+Xlimit;
  else
   Xlimit=256*int(ch)+Xlimit;
  fi.seekg(22);
  fi.get(ch);
  Ylimit=int(ch);
  if(Ylimit<0)
   Ylimit=256+Ylimit;
```

```
  fi.get(ch);
  if(int(ch)<0)
   Ylimit=256*(256+int(ch))+Ylimit;
  else
   Ylimit=256*int(ch)+Ylimit;
  int size=(Xlimit+1)/2,y=Ylimit-1,num;
  blankspace=filesize-118-Ylimit*size;
  blankspace/=(Ylimit-1);
  i=0;
  fi.seekg(118);
  fi.get(ch);
  if(use==1)
  {
    CANVASEX=CANVASSX+Xlimit+1;
    CANVASEY=CANVASSY+Ylimit+1;
    canvasresize();
  }
  while(fi)
  {
    for(int j=0;j<size;j++)
    {
      num=int(ch);
      if(num<0)
      num+=256;
      colo(num);
      if(type==1)
      {
        if(shiftx+(j*2)<CANVASEX&&shifty+y<CANVASEY)
         putpixel(shiftx+(j*2),shifty+y,colo1);
        if(shiftx+(j*2)<CANVASEX-1&&shifty+y<CANVASEY)
         putpixel(shiftx+1+(j*2),shifty+y,colo2);
      }
      else
      {
        putpixel(shiftx+(j*2),shifty+y,colo1);
        putpixel(shiftx+1+(j*2),shifty+y,colo2);
      }
      fi.get(ch);
    }
    i++;
    y--;
    fi.seekg(118+(size+blankspace)*i);
    fi.get(ch);
  }
  return 0;
}
//************************************************
void create()
{
     cleardevice();
     fstream fout("emp.dat",ios::binary|ios::app);
     //opening the master file in output mode for writing
```

```
     fish:
    ob1.getdata();
    fout.write((char*)&ob1,sizeof(ob1));
    cleardevice();
    bmp("save.bmp",170,35,300,225);
    delay(200);
    cleardevice();
    bmp("YN.bmp",20,0);
    click();
    if ((x>=162&&x<=271)&&(y>=188&&y<=283))
    goto fish;
    else if ((x>=310&&x<=431)&&(y>=173&&y<=289))
    {
        fout.close();
        void mm();
        mm();
  }
    }
void display()
{
    cleardevice();
    gotoxy(25,1);
    //loop to display data
      fstream fin("emp.dat",ios::in|ios::binary);
      gotoxy(25,1);
      cout<<"\nEnter the Employee number whose data is to be displayed : ";
      int empno2,flag=0;
      cin>>empno2;
      while(fin.read((char*)&ob1,sizeof(ob1)))
         {
              if(ob1.Return_empno()==empno2)
                {  ob1.dispdata();
                   flag=1;
                   }
                }
         if (flag==0)
         {
           cout<<"\nEmployee you were"
           <<" searching for is not found please try again ";
           delay(200);
           }
         fin.close();
         getch();
         void mm();
         mm();
       }
void Delete()
{
   cleardevice();
   fstream fout("temp.dat",ios::binary|ios::out);
   fstream fin("emp.dat",ios::binary|ios::in);
   a1:
```

```
  gotoxy(25,1);
  cout<<"\nEnter the Employee number whose data is to be deleted : ";
  int empno2,flag=0;
  cin>>empno2;
while(fin.read((char*)&ob1,sizeof(ob1)))
 {
   if(ob1.Return_empno()!=empno2)
   fout.write((char*)&ob1,sizeof(ob1));
  else
  flag=1;
  }
  fin.close();
  fout.close();
  if(flag==0)
     {
        cout<<"\nEmployee not found\nEnter again ";
        delay(500);
        cleardevice();
        goto a1;
   }

 else
{
   ob1.dispdata();
   cout<<"\n\nARE YOU SURE YOU WANT TO DELETE THIS DATA (y/n) : ";
   char rep;
   cin>>rep;
   if(rep=='y'||rep=='Y')
   {
      remove("emp.dat");
      rename("temp.dat","emp.dat");
      cleardevice();
      bmp("del.bmp",0,-50);
      getch();
      void mm();
      mm();
    }
   else
   {
      cout<<"n\n\t\tDATA WILL NOT BE DELETED ";
      delay(700);
      void mm();
      mm();
  }
 }
}
void modify()
{
   char rep;
   int flag=0;
   fstream fin("emp.dat",ios::binary|ios::in);
   fstream fout("temp.dat",ios::binary|ios::out);
```

```
   a2:
   cleardevice();
   gotoxy(25,1);
   cout<<"\nEnter the Employee number"
   <<" whose data is to be modified : ";
   int empno2;
   cin>>empno2;
   while(fin.read((char*)&ob1,sizeof(ob1)))
     {
        if(ob1.Return_empno()!=empno2)
          fout.write((char*)&ob1,sizeof(ob1));
        else
          flag=1;
        }
     if(flag==0)
       {
        cout<<"\nEmployee not found \nEnter again ";
        delay(500);
        goto a2;
             }

        else
        {
           ob1.getdata();
           fout.write((char*)&ob1,sizeof(ob1));
           }
        fin.close();
        fout.close();
        remove("emp.dat");
        rename("temp.dat","emp.dat");
        cleardevice();
        bmp("save.bmp",170,35,300,225);
        delay(200);
        void mm();
        mm();

     }
void attendance()
{
   cleardevice();
   fstream fin("emp.dat",ios::binary|ios::in);
   fstream fin1("att.dat",ios::binary|ios::in);
   gotoxy(25,1);
   cleardevice();
cout<<"\nEnter the employee number of the Employee you want to search : ";
   int emp,flag=0, flag1=0;
   cin>>emp;
while(fin.read((char*)&ob1,sizeof(ob1)))
{

 if(ob1.Return_empno()==emp)
 {
```

```
  flag=1;
  break;
  }}
  while(fin1.read((char*)&att,sizeof(att)))
{
   if(att.Empno==emp)
    {
    cout<<"\n Employee name   : "<<ob1.Return_name();
    cout<<"\n Department Name : "<<ob1.Department;
    cout<<"\n Designation     : "<<ob1.Designation;
    att.Adisplaydata();
    flag1=1;
    break;
    }
}
if (flag==0 || flag1==0)

 cout<<"\n\n\t\t Employee you are looking for is"
 <<" not found\nPlease try again  ";
  getch();
  void mm();
  mm();
 }
void dept()
{
   cleardevice();
   fstream fin("emp.dat",ios::binary|ios::in);
   a3:
   gotoxy(25,1);
cout<<"\nEnter the Department of the Employee you want to search : ";
   char dept2[20];
   gets(dept2);
while(fin.read((char*)&ob1,sizeof(ob1)))
{
if(strcmpi(ob1.Department,dept2)==0)
ob1.dispdata();
else
 {
 cout<<"\n\n\t\t Employee you are looking for is"
 <<" not found\nPlease try again  ";
 delay(800);
 cleardevice();
 goto a3;
  }
 }
getch();
void mm();
mm();

 }
void desig()
{
```

```
    cleardevice();
    fstream fin("emp.dat",ios::binary|ios::in);
    a4:
    gotoxy(25,1);
cout<<"\nEnter the Designation of the Employee you want to search : ";
    char desig2[20];
    gets(desig2);
while(fin.read((char*)&ob1,sizeof(ob1)))
{
  if(strcmpi(ob1.Designation,desig2)==0)
  ob1.dispdata();
  else
  {
  cout<<"\n\n\t\t Employee you are looking for"
      <<" is not found\nPlease try again  ";
  delay(800);
  cleardevice();
  goto a4;
     }
    }
getch();
void mm();
mm();
  }
void employee::report()
{
  void loading();
  loading();
  cleardevice();
//to draw the table
  rectangle(0,0,630,300);
  line(0,50,630,50);
  line(0,90,630,90);
  gotoxy(30,2);
  cout<<" PF STATEMENT ";
  gotoxy(2,5);
  cout<<"EmpNo";
  gotoxy(12,5);
  cout<<"NAME";
  gotoxy(20,5);
  cout<<"POST";
  gotoxy(30,5);
  cout<<"BASIC";
  gotoxy(40,5);
  cout<<"GROSS PAY";
  gotoxy(52,5);
  cout<<"PF";
  gotoxy(58,5);
  cout<<"DEDUCTION";
  gotoxy(70,5);
  cout<<"NETPAY";
  line(50,90,50,300); //to draw verical line(x1,y1,x2,y2)
```

```
 line(140,90,140,300);
 line(220,90,220,300);
 line(290,90,290,300);
 line(390,90,390,300);
 line(440,90,440,300);
 line(540,90,540,300);
 int k=8;
 fstream fin("emp.dat",ios::binary|ios::in);
if(!fin)
 {
  cout<<"\n\nFile Not Found...\nProgram Terminated!";
  delay(100);
  exit(0);
  }
 fin.seekg(0);
 while(fin.read((char*)&ob1,sizeof(ob1)))
 {
   gotoxy(3,k);
   cout<<ob1.empno;
   gotoxy(10,k);
   cout<<ob1.Name;
   gotoxy(20,k);
   cout<<ob1.Designation;
   gotoxy(30,k);
   cout<<ob1.Basic;
   gotoxy(40,k);
   cout<<ob1.Gross;
   gotoxy(50,k);
   cout<<ob1.PF;
   gotoxy(58,k);
   cout<<ob1.Tax;
   gotoxy(70,k);
   cout<<ob1.Net;
   k=k+1;
       }
   fin.close();
   getch();
   void mm();
   mm();

  }

void employee::Salary()
{
   fstream fin("emp.dat",ios::in|ios::binary);
   fstream fin2("att.dat",ios::in|ios::binary);
   loading();
   cleardevice();
  int eno,flag=0;
  gotoxy(25,1);
  cout<<"\n Enter employee number to be searched :";
  cin>>eno;
```

```
  fin.seekg(0);//to read from zeroth byte
//reading from both the files
  while(fin2.read((char *)&att,sizeof(att)))
  {
      if(att.Empno==eno)
      {
      flag=1;
      break;
      }
  }
  int l=att.LWP;
while(fin.read((char*)&ob1,sizeof(ob1)))
{
 if(ob1.empno==eno)
 {
   gotoxy(15,4);
   cout<<"\n\t*********************************************";
   cout<<"\n\t        STARK INDUSTRIES PAYSLIP          ";
   cout<<"\n\t*********************************************";
   cout<<"\n\tEMPLOYEE NUMBER      :"<<ob1.empno;
   cout<<"\n\tNAME OF EMPLOYEE     :"<<ob1.Name;
   cout<<"\n\tEMPLOYEE DESIGNATION :"<<ob1.Designation;
   cout<<"\n\tCL Avail :"<<att.casleaveavail
       <<"\n\tEL Avail :" <<att.earnleaveavail
       <<"\n\tLWP      :"<<att.LWP;
   if(l>=0)
  {
   ob1.Basic=ob1.Basic/30*(30-l);
   ob1.Hra=ob1.Hra/30*(30-l);
   ob1.DA=ob1.DA/30*(30-l);
   ob1.Gross=ob1.Basic+ob1.Hra+ob1.DA;
   ob1.PF=ob1.PF/30*(30-l);
   ob1.Net=ob1.Gross-ob1.PF;
 }
   cout<<"\n\tBASIC SALARY        :"<<ob1.Basic;
   cout<<"\n\n\tALLOWANCES : ";
   cout<<"\tHRA: "<<ob1.Hra<<"\n\t\t\tDA: "<<ob1.DA;
   cout<<"\n\n\tGROSS PAY          :"<<ob1.Gross;
   cout<<"\n\tDEDUCTIONS : ";
   cout<<"\n\t\t\tPF: "<<ob1.PF;
   cout<<"\n\t\t\tTAX:"<<ob1.Tax;
   cout<<"\n\n\tNET PAY            :"<<ob1.Net;
   cout<<"\n\t*********************************************";
   flag=1;
   getch();
 }
}
if(flag==0)
{
 cleardevice();
 cout<<"\n\tNo such employee exists in our database ....";
 getch();
```

```
        }
 fin.close();
 void mm();
 mm();

   }
void attendance::Agetdata()
{

  int flag=0;
  starts:
  cleardevice();
  fstream fin("emp.dat",ios::in|ios::binary);
  gotoxy(25,1);
  cout<<"\nENTER THE Employee NUMBER : ";
  cin>>Empno;
  while(fin.read((char*)&ob1,sizeof(ob1)))
    {
       if(Empno==ob1.Return_empno())
         {

              cout<<"\nENTER Number of Casual Leaves : ";
              cin>>casleave;
              cout<<"\nENTER Number of Earned Leaves :  ";
              cin>>earnleave;
              earnleaveavail=casleaveavail=LWP=0;
               flag=1;
              break;
                     }
 }
   fin.close();
   if (flag==0)
   {
     cout<<"the employee number you "
     <<"entered is not in our database please enter again ";
     delay(400);
     void mm();
     mm(); //if wrong employee
    //number is input then will ask to enter again
   }
}
void attendance::Adisplaydata()
{
    gotoxy(25,1);
    cout<<"\nEmployee number is : "<<Empno;
    cout<<"\nThe number of casual leaves :"<<casleave;
    cout<<"\nThe number of earned leaves :"<<earnleave;
    cout<<"\nThe number of casual leaves Availed : "<<casleaveavail;
    cout<<"\nThe number of earned leave Availed :"<<earnleaveavail;
    cout<<"\nLEAVE WITHOUT PAY : "<<LWP;   }
void monthly()
{
```

```
    int Empno,flag=0;
    cleardevice();
    fstream fin("emp.dat",ios::in|ios::binary);
    fstream fin1("att.dat",ios::in|ios::out|ios::binary);
    sun:
    gotoxy(25,1);
    cout<<"\nENTER the Employee NO: ";
    cin>>Empno;
      while(fin.read((char*)&ob1,sizeof(ob1)))
         {
           if(Empno==ob1.Return_empno())
            {
               flag=1;
               break;
             }
                    }
        if (flag==0)
      { cout<<"\nyou have not entered"
           <<" a valid employee number"
           <<" please try again ";
       delay(200);
       goto sun;
                  }
            cleardevice();
            flag=0;
 while(fin1.read((char*)&att,sizeof(att)))
  {
       if(Empno==att.Empno)
          {
            flag=1;
            break;
                }
             }
       if (flag==0)
        { cout<<"\nyou have not entered a"
             <<"valid employee number"
             <<"please try again ";
              delay(200);
              goto sun;
                  }
            cleardevice();

int leave_taken;
cout<<"\nEnter No of CL leaves taken this month: ";
cin>>att.casleaveavail;
if (att.casleave<att.casleaveavail)
{
cout<<"\n\n \t\t Sorry these many cl leaves are not available";
cout<<"\n\n \t\t You can take "<<att.casleave<<" leaves";
att.casleaveavail=att.casleave;
att.casleave=0;
}
```

```
else
{
att.casleave-=att.casleaveavail;
}
cout<<"\nEnter No of EL leaves taken this month: ";
cin>>att.earnleaveavail;
if (att.earnleave<att.earnleaveavail)
{
cout<<"\n\n \t\t Sorry these many el leaves are not available";
cout<<"\n\n \t\t You can take "<<att.earnleave<<" leaves";
att.earnleaveavail=att.earnleave;
att.earnleave=0;
}
else
{
att.earnleave-=att.earnleaveavail;
}
cout<<"\nEnter no of Leave Without Pay(if any and enter 0 if no ): ";
cin>>att.LWP;

long offset=fin1.tellg()-sizeof(att); //to go to previous record
fin1.seekp(offset);
fin1.write((char *)&att,sizeof(att));
cout<<"\n Leave updated";
fin1.close();
fin.close();
getch();
void mm();
mm();
          }
void am()//attendance management
{
 cleardevice();
 gotoxy(25,1);
 cout<<"\n1.Add attendance records\n2.Display attendance"
 <<" records\n3.UPDATE attendance records";
    cout<<"\nEnter your choice(1-3): ";
    int choice;
    cin>>choice;
switch(choice)
{
   case 1:
  {
    fstream fout("att.dat",ios::binary|ios::app);
    fish1:
    att.Agetdata();
    fout.write((char*)&att,sizeof(att));
    cleardevice();
    bmp("save.bmp",170,35,300,225);
    delay(200);
    cleardevice();
    bmp("YN.bmp",20,0);
```

```
    click();
    if ((x>=162&&x<=271)&&(y>=188&&y<=283))
    goto fish1;
    else if ((x>=310&&x<=431)&&(y>=173&&y<=289))
    {
        fout.close();
        void mm();
        mm();
         }
           }
     break;
  case 2:
  {
        cleardevice();
        gotoxy(25,1);
    //loop to display data
     fstream fin("att.dat",ios::in|ios::binary);
     int Empno,flag=0;
cout<<"enter the employee number you want to display data for :";
cin>>Empno;
 while(fin.read((char*)&att,sizeof(att)))
    {
      if(Empno==att.Empno)
        {
            att.Adisplaydata();
                flag=1;
          }
}
if(flag==0)
cout<<"employee does not exist ";
     fin.close();
     getch();
     void mm();
     mm();
   }
      break;
 case 3:
  {
    cleardevice();
    bmp("R.bmp",30,-40);
    moon:
    click();
    if ((x>=127&&x<=281)&&(y>=102&&y<=293))
     {
          //"YEARLY UPDATE"
          loading();
          cleardevice();
          int flag=0;
           starts:
           gotoxy(25,1);
           fstream fin("emp.dat",ios::in|ios::binary);
           fstream fin1("att.dat",ios::in|ios::out|ios::binary);
```

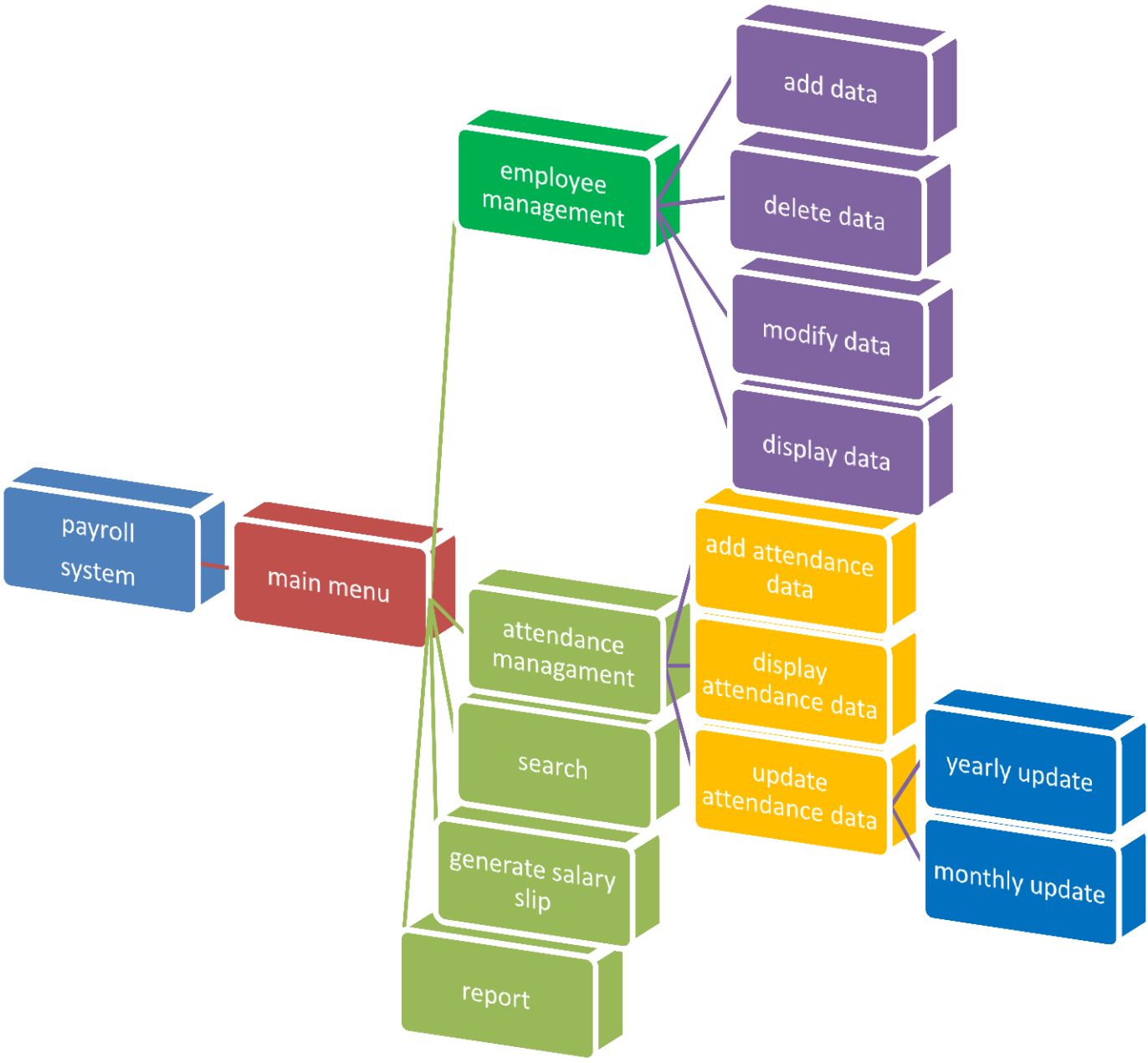
```
        cout<<"\nENTER THE Employee NUMBER : ";
        cin>>att.Empno;
      while(fin.read((char*)&ob1,sizeof(ob1)))
       {
        if(att.Empno==ob1.Return_empno())
          {
             cout<<"\nENTER Number of Casual Leaves : ";
             cin>>att.casleave;
             cout<<"\nENTER Number of Earned Leaves :  ";
             cin>>att.earnleave;
             att.earnleaveavail=att.casleaveavail=att.LWP=0;
             fin1.write((char*)&att,sizeof(att));
              flag=1;
             break;
        }
 }
   fin.close();
   fin1.close();
   if (flag==0)
   {
     cout<<"\nthe employee number you "
     <<"entered is not in our database please enter again ";
     delay(200);
     goto starts; //if wrong employee
    //number is input then will ask to enter again

  }         }
   else if ((x>=367&&x<=533)&&(y>=103&&y<=287))
     {
        //MONTHLY UPDATE
        monthly();
      }
   else if ((x>=563&&x<=593)&&(y>=246&&y<=340))
   {
      exit(0);
     }
   else
   goto moon;
     }
     break;
   }//end of switch
 }
void ded()
{
    cleardevice();
    bmp("ded.bmp",70,30);
    island:
    click();
    if ((x>=108&&x<=233)&&(y>=104&&y<=162))
    {
       create();
    }
```

```
    else if ((x>=349&&x<=524)&&(y>=151&&y<=165))
    {
      Delete();
      }
    else if ((x>=107&&x<=281)&&(y>=184&&y<=320))
    {
       display();
    }
    else if ((x>=349&&x<=541)&&(y>=188&&y<=321))
    {
        modify();
    }
    else if ((x>=511&&x<=540)&&(y>=310&&y<=490))
    {
      exit(0);
      }
      else
      goto island;
    }
void search()
{
   cleardevice();
   bmp("P.bmp",0,0);
   batman:
   click();
   if ((x>=104&&x<=178)&&(y>=101&&y<=324))
   {
     dept();

   }
   else if ((x>=236&&x<=394)&&(y>=123&&y<=320))
   {
    attendance();
    }
    else if ((x>=457&&x<=596)&&(y>=125&&y<=327))
    {
      desig();
      }
      else if ((x>=547&&x<=573)&&(y>=151&&y<=561))
      {
        exit(0);
        }
        else
        goto batman;
   }
void mm()
{
  cleardevice();
  setbkcolor(BLACK);
  bmp("f.bmp",70,-20);
  star:
  callmouse();
```

```
   click();
if((x>=409&&x<=500)&&(y>=196&&y<=339))
{
    ob1.report();
      }
else if ((x>=107&&x<=240)&&(y>=148&&y<=181))
{
  ded();//display employee data ded.bmp
  }
else if ((x>=291&&x<=398)&&(y>=149&&y<=178))
{
  am();
    }
else if ((x>=453&&x<=519)&&(y>=149&&y<=174))
{    search();
 }
else if ((x>=518&&x<=544)&&(y>=314&&y<=333))
 { exit(0);
}
else if ((x>=110&&x<=279)&&(y>=208&&y<=338))
{  ob1.Salary();
 }
else
goto star;
    }
void main()
{
          /* select a driver and mode that supports */
   /* multiple background colors.            */
   int gdriver = EGA, gmode = EGAHI, errorcode;
   int bkcol,  x, y;
   char msg[80];

   /* initialize graphics and local variables */
   initgraph(&gdriver, &gmode, "");

   /* read result of initialization */
   errorcode = graphresult();
   if (errorcode != grOk)  /* an error occurred */
   {
      printf("Graphics error: %s\n", grapherrormsg(errorcode));
      printf("Press any key to halt:");
      getch();
      exit(1); /* terminate with an error code */
   }

   /* maximum color index supported */
  // maxcolor = getmaxcolor();

   /* for centering text messages */
   settextjustify(CENTER_TEXT, CENTER_TEXT);
bmp("c.bmp",50,-35,640,480);
```

```
delay(3000);
cleardevice();
char pass[30],ch;
int i=-1;
setbkcolor(BLACK);

 bmp("ps.bmp",20,0);
 do
 {
    ch=getch();
    if (ch==13)//enter key
    break;
    if(ch==8)//backspace key
    {   if(i>=0)
        i--;
        cout<<'\0';
        }
    else
    pass[++i]=ch;
    gotoxy(34,17.79);
    for(int x=0;x<=i;x++)
    cout<<"*";
    }while(1);
    pass[++i]='\0';
    if(strcmp(pass,"hello")==0)
    {
           cleardevice();
           bmp("success.bmp",20,0);
           delay(100);
           cleardevice();
           loading();
           mm();
           }
           else
       {    cleardevice();
            gotoxy(35,12);
            setbkcolor(14);
            cout<<"\nYOU ENTERED THE WRONG PASSWORD";
            delay(1000);
            cleardevice();
            loading();
            cleardevice();
for(int h=5;h>0;h--)
{         gotoxy(35,12);
          setbkcolor(h);
          cout<<"PROGRAM TERMINATING IN "<<h<<" SECONDS";
 delay(1000);
 cleardevice();
 }        exit(0);
     } //end of else of password
 getch();
}//end of main
```

LAYOUT


payroll system
main menu
employee management
add data
delete data
modify data
display data
attendance managament
add attendance data
display attendance data
update attendance data
yearly update
monthly update
search
generate salary slip
report

BIBLIOGRAPHY

1. C++ by Robert Lafore

2. Computer Science by Sumita Arora

3. Internet

Compilation Report