DESCRIPTION:

EMPLOYEE RECORD SYSTEM:

In this project Employee record system, It is a concept of creating a record for employee and to modify the working time ,salary etc.., . Here include various data member and member function to perform the record system such as:

1. Create the record.

2. Display the employee details.

3.Search the employee datails.

4.To Modify the details of the employee.

First, In the class **EMPDETAILS** get the basic detals of the employee such as name, idno , age , gender etc…

In the class **CREATEREC** creating the particular employee details and allocating the work for the day.

Then in class **DISPEMP** get the name of the employee from user and displaying the record of that employee in a manner

In the class **SEARCHEMP** get the idno from the user and search the employee using the idno , if the employee found then check the employee is regular or partime worker by hours of working (if working hours is greater than 6) then the worker is regular worker or he is partime worker or if the employee is not found then print “Employee not found”.

In class **MODIFYEMP** get the information from the user and modify the employee’s details by changing their hours of working and their idno and salary and then finally display the modified details of the Employee.

In main function by using switch case call every function to execute and end the program.

CONCEPTS USED IN EMPLOYEE RECORD SYSTEM PROGRAM:

1. HIERARCHIAL INHERITANCE:

Hierarchial Inheritance is a type of inheritance where many subclasses inherit from single class .

2.CLASSES AND OBJECT:

In C++ programming, a Class is a fundamental block of a program that has its own set of methods and variables. You can access these methods and variables by creating an object or the instance of the class.

3.STATIC MEMBER DATA:

Static data members are class members that are declared using the static keyword. There is only one copy of the static data member in the class, even if there are many class objects. This is because all the objects share the static data member.

4. CONSTRUCTOR:

A constructor in C++ is a special 'MEMBER FUNCTION' having the same name as that of its class which is used to initialize some valid values to the data members of an object. It is executed automatically whenever an object of a class is created.

5. AUTO STORAGE CLASS:

The auto keyword is a simple way to declare a variable that has a complicated type. For example, you can use auto to declare a variable where the initialization expression involves templates, pointers to functions, or pointers to members.

4.INLINE FUNCTION:

Inline function in C++ is an enhancement feature that improves the execution time and speed of the program.

7.SCOPE RESOLUTION:

The Scope Resolution Operator is also called as simpler terms, the double colon, is a token that allows access to static, constant, and overridden properties or methods of a class. When referencing these items from outside the class definition, use the name of the class.

PROGRAM:

```
#include<iostream.h>
#include<conio.h>
#include<iomanip.h>
class empdetails
{
public:
char name[100];
char dob[50];
char sex[50];
static long idno;
long salary;
int age;
int year,hours;
public:
inline void getname()
{
cout<<"Name of the Employee:"<<endl;
cin>>name;
}
public:
void empid()
{
cout<<"Employee-ID:"<<endl;
cin>>idno;
}
public:
void datebirth()
```

```
{
cout<<"Date of birth:"<<endl;
cin>>dob;
}
public:
void getage()
{
cout<<"Age of the Employee:"<<endl;
cin>>age;
}
public:
void getsalary()
{
cout<<"Salary of the Employee:"<<endl;
cin>>salary;
}
public:
void getyears()
{
cout<<"No.of.Years Working:"<<endl;
cin>>year;
 }
public:
void gethours()
{
 cout<<"Hours of Working:"<<endl;
 cin>>hours;
 }
 public:
```

```
void getsex();
public:
void getempid()
{
cout<<idno<<endl;
 }
public:
void nameemp()
{
cout<<name<<endl;
}
public:
void emp_hours()
{
cout<<hours<<endl;
}
public:
void salaryemp()
{
cout<<salary<<endl;
}
};
void empdetails::getsex()
{
cout<<"Gender:"<<endl;
cin>>sex;
}
long empdetails::idno;
class create:protected empdetails
```

```
{
public:
void createrec()
{
cout<<"    Create The Employee Details    "<<endl;
cout<<endl;
getname();
empid();
datebirth();
getsex();
getage();
getsalary();
gethours();
}
};
class dispemp:protected empdetails
{
public:

void display()
{
cout<<"Information Of Employee "<<endl;
getname();
empid();
getsalary();
gethours();
cout<<"   Displaying the records of employee:   "<<endl;
cout<<endl;
cout<<"NAME:  "<<name<<endl;
```

```
cout<<"EMPLOYEE ID: "<<idno<<endl;
cout<<"SALARY: "<<salary<<endl;
cout<<"Hours of Working:"<<hours<<endl ;
}
};
class searchemp:public empdetails
{
public:
void search()
{
int ID;
cout<<"   To Search the Employee   "<<endl;
cout<<endl;
empid();
cout<<"Enter the id no. to search the employee:"<<endl;
cin>>ID;
if(ID==idno)
{
cout<<"The searched employee is found "<<endl;
gethours();
cout<<"Type Of Searched Employee"<<endl;
cout<<endl;
if(hours>6)
{
cout<<"The employee is regular worker"<<endl;
}
else
{
cout<<"The employee is partime worker"<<endl;
} }
else
{
cout<<"Employee not found"<<endl;
} } };
class modifyemp:protected empdetails
{
public:
void modify()
{
cout<<"Employee's Details To Modify are"<<endl;
cout<<endl;
getname();
empid();
gethours();
getsalary();
cout<<" MODIFYING THE EMPLOYEE'S DETAILS  "<<endl;
cout<<endl;
int new_id;
int new_hours;
cout<<"Modified Id no:"<<endl;
cin>>new_id;
cout<<"Modified Working Hours Of The Employee:"<<endl;
cin>>new_hours;
cout<<"NAME : "<<setw(10)<<name<<endl<<"ID_NO:
"<<setw(10)<<new_id<<endl<<"Hours:
"<<setw(10)<<new_hours<<endl<<"Salary:"<<setw(10)<<salary<<endl;
} };
void main()
```

```
{
int i=0;
searchemp s[1];
dispemp d[1];
create c[1];
modifyemp m[1];
int choice;
clrscr();
cout<< "****** EMPLOYEE RECORD SYSTEM ******"<<endl;
cout<<endl;
cout<<"  PRESS 1 TO CREATE THE RECORD  "<<endl;
cout<<endl;
cout<<" PRESS 2 TO DISPLAY THE RECORD  "<<endl;
cout<<endl;
cout<<"  PRESS 3 TO SEARCH THE RECORD   "<<endl;
cout<<endl;
cout<<" PRESS 4 TO MODIFY THE RECORD  "<<endl;
cout<<endl;
cout<<" PRESS 5 TO EXIT"<<endl;
cout<<endl;
do
{
cout<<"Enter the choice:"<<endl;
cin>>choice;
switch(choice)
{
case 1:
c[i].createrec();
cout<<endl;
break;
case 2:

d[i].display();
cout<<endl;

break;
case 3:
s[i].search();

cout<<endl;
break;

case 4:
m[i].modify();
cout<<endl;
break;

} }
while(choice!=5);
cout<<"Thank you...."<<endl;
getch();

}
```