**Theory Assignment-3**

**Code:-**

//Harsh Bamotra AC-1216

//Program to create an employee class and show polymorphism

#include <iostream>

using namespace std;

// \*\*\*\*\*\*\*\* Defining base class employee \*\*\*\*\*\*\*\*

class employee

{

protected: //Defining protected member

string E\_name , Department; //used protected and public members because the

int E\_number , age; //private members are not inherited in the derived classes

public: //Defining public member

employee() //defining constructor

{

cout << "Enter the employee name ::";

getline(cin , E\_name);

cout << "Enter your age::";

cin >> age;

cin.ignore();

cout << "Enter your department::"; //defined as public so that

getline(cin , Department); //they can be accessible

cout << "Enter your employee number::";

cin >> E\_number;

}

virtual void computeSalary()=0; //defining virtual functions

virtual void display()=0;

}

//\*\*\*\*\*\*\* Defining derived class regularEmployee \*\*\*\*\*\*\*

class regularEmployee : public employee

{ //inherited as public so that the members can be accessible

//by the bass class pointer

private: //Defining private members

float Basic\_pay , HRA , TA , DA , Gross\_sal , tax , Net\_sal; //defined as private so that they

//can't be accessed directly

public: //Defining public members

regularEmployee() //Defining constructor

{

cout << "Enter your basic pay::"; //defined as public so that they can

cin >> Basic\_pay; //be accessible in the main function

}

void computeSalary() //overriding function

{

TA=10000;

HRA=0.3\*Basic\_pay;

DA=0.8\*(Basic\_pay+HRA+TA);

Gross\_sal=Basic\_pay+HRA+TA+DA;

tax=0.3\*Gross\_sal;

Net\_sal=Gross\_sal-tax;

}

void display() //overriding function

{

cout << "Name::" << E\_name << endl;

cout << "Department::" << Department << endl;

cout << "Employee No. ::" << E\_number << endl;

cout << "Age ::" << age << endl;

cout << "The net salary of the employee::" << Net\_sal << endl << endl;

}

};

//\*\*\*\*\*\* Defining derived class conEemployee \*\*\*\*\*\*

class conEmployee : public employee //inherited as public so that the members can be accessible

{ //by the bass class pointer //defining private members

private:

float Hourly\_Rate , No\_Hours , Gross\_sal , tax , Net\_sal; //defined as private so that they

//can't be accessed directly

public: //Defining public members

conEmployee() //Defining constructor

{

cout << "Enter the number of work hours::";

cin >> No\_Hours;

cout << "Enter the hour rate::"; //defined as public so that they

cin >> Hourly\_Rate; //can be accessible in the main function

}

void computeSalary() //overriding function

{

Gross\_sal=Hourly\_Rate\*No\_Hours;

tax=0.1\*Gross\_sal;

Net\_sal=Gross\_sal-tax;

}

void display() //overriding function

{

cout << "Name::" << E\_name << endl;

cout << "Department::" << Department << endl;

cout << "Employee No. ::" << E\_number << endl;

cout << "Age ::" << age << endl;

cout << "The net salary of the employee::" << Net\_sal << endl << endl;

}

};

int main()

{

int ch;

employee \*em; //defining employee class pointer

//printing menu

cout << "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* MENU \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" << endl;

cout << "1. Regular employee" << endl << "2. Contractual employee" << endl;

cout << "Enter your choice(1 or 2)::";

cin >> ch;

cin.ignore();

if(ch==1)

{

cout << endl;

cout << "\*\*\*\*\*\*\*\*\*\*\*\*\* ENTER YOUR DETAILS \*\*\*\*\*\*\*\*\*\*\*\*\*" << endl;

regularEmployee emp1; //defining regularEmployee class object

em = &emp1; // pointing the pointer to object

cout << "\*\*\*\*\*\* Details of the regular employee \*\*\*\*\*\*" << endl;

em -> computeSalary();

em -> display(); //printing the data

}

else if(ch==2)

{

cout << endl;

cout << "\*\*\*\*\*\*\*\*\*\*\*\*\* ENTER YOUR DETAILS \*\*\*\*\*\*\*\*\*\*\*\*\*" << endl;

conEmployee emp2; //defining conEmployee class object

em = &emp2; // pointing the pointer to object

cout << "\*\*\*\*\*\*\* Details of the contractual employee \*\*\*\*\*\*" << endl;

em -> computeSalary();

em -> display(); //printing the data

}

else

{

cout << "Wrong Input !!! Exiting";

}

return 0;

}

**Output:-**

**1.Regular Employee**

A screenshot of a computer

Description automatically generated

**2. Contractual Employee**

**A picture containing text, monitor, electronics, screenshot

Description automatically generated**

**3. Handling Exception**

**A picture containing text, monitor, screen, computer

Description automatically generated**

**Harsh Bamotra**

**AC-1216**