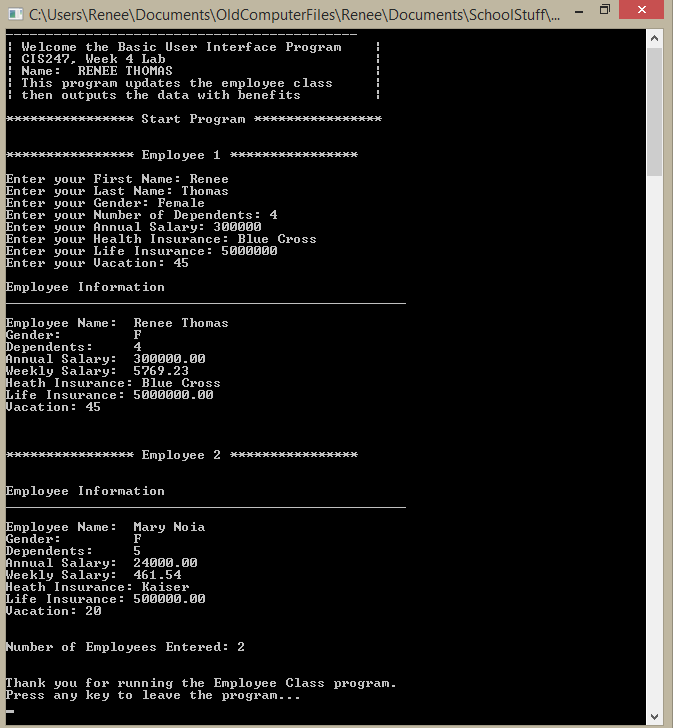
**Screen Shot:**



**Code:**

// ---------------------------------------------------------------------------

//Program Header

//Program Name: Employee Class Integration

//Programmer: Renee Thomas

//CIS247, Week 4 Lab

//Program Description: Update Employee Class to inclued a Benefits class

// ---------------------------------------------------------------------------

# include <iostream>

# include <string>

# include <iomanip>

using namespace std;

// create prototypes for functions

void DisplayApplicationInformation();

void DisplayDivider(string);

string getInput(string);

void TerminateApplication();

// create Benefits Class

class Benefits{

private:

string healthInsurance;

double lifeInsurance;

int vacation;

public:

Benefits();

Benefits(string hI, double lI, int vac);

void displayBenefits();

string getHealthInsurance();

void setHealthInsurance(string hI);

double getLifeInsurance();

void setLifeInsurance(double lI);

int getVacation();

void setVacation(int vac);

};

// create iEmployee class

class iEmployee{

public:

virtual double calculatePay(double) = 0;// make sure to make virtual method EXACTLY the same as the method you want in the other classes including parameters!

};

// create Employee Class

class Employee : public iEmployee

{

private:

string firstName;

string lastName;

char gender;

int dependents;

double annualSalary;

static int numEmployees;

public:

Employee();

Employee(string fn, string ln, char gen, int dep, double sal, Benefits bens\_in); // don't forget to use the class name as the data type!

double calculatePay(double sal);

void displayEmployee();

string getFirstName();

void setFirstName(string fn);

string getLastName();

void setLastName(string ln);

char getGender();

void setGender(char gen);

int getDependents();

void setDependents(int dep);

double getAnnualSalary();

void setAnnualSalary(double sal);

static int getNumEmployees();

void setDependents(string dep);

void setAnnualSalary(string sal);

Benefits bens;// create an instance of Benefits inside Employee class

};

int main(){

// call function for header information

DisplayApplicationInformation();

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Start Program\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// call function for divider for start program

DisplayDivider("Start Program");

DisplayDivider("Employee 1");

// create default employee to set variables below

Employee emp1;

//set variables

emp1.setFirstName(getInput("First Name: "));

emp1.setLastName(getInput("Last Name: "));

emp1.setGender(getInput("Gender: ")[0]);

emp1.setDependents(getInput("Number of Dependents: "));

emp1.setAnnualSalary(getInput("Annual Salary: "));

emp1.bens.setHealthInsurance(getInput("Health Insurance: "));

// create variable so user input can be changed to correct data type

string input;

// get Life insurance from user

input = getInput("Life Insurance: ");// use the input variable to store the users string input

double lI = atof(input.c\_str());// change user input to double and add to lI variable

emp1.bens.setLifeInsurance(lI);// set instance of bens/LiveInsurance inside the Employee class to lI

//get vacation from user

input = getInput("Vacation: ");// use the input variable to store the users string input

int vac = atoi(input.c\_str());// change the string input to an integer and place inside of the vac variable

emp1.bens.setVacation(vac);// set instance of bens/vacation inside the Employee class to vac

// call class function to display employee information

emp1.displayEmployee();

DisplayDivider("Employee 2");

Benefits benefits1("Kaiser", 500000, 20);

// create new employee per iLab3 instructions passing in all the values

Employee emp2("Mary", "Noia", 'F', 5, 24000.0, benefits1);

// call class function to display employee information

emp2.displayEmployee();

// call the class static function to get number of employees created

cout<<"Number of Employees Entered: "<<Employee::getNumEmployees()<<endl;

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*End Program\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// call function to leave program

TerminateApplication();

// keep page open until user hits any key

cin.ignore();

return 0;

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Benefits Class Functions\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Benefits::Benefits(){

healthInsurance = "Not provided";

lifeInsurance = 0.0;

vacation = 14;

}

Benefits::Benefits(string hI, double lI, int vac){

healthInsurance = hI;

lifeInsurance = lI;

vacation = vac;

}

void Benefits::displayBenefits(){

cout<<"Heath Insurance: "<<healthInsurance<<endl;

cout<<"Life Insurance: "<<lifeInsurance<<endl;

cout<<"Vacation: "<<vacation<<endl;

}

string Benefits::getHealthInsurance(){return healthInsurance;}

void Benefits::setHealthInsurance(string hI){healthInsurance = hI;}

double Benefits::getLifeInsurance(){return lifeInsurance;}

void Benefits::setLifeInsurance(double lI){lifeInsurance = lI;}

int Benefits::getVacation(){return vacation;}

void Benefits::setVacation(int vac){vacation = vac;}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Employee Class Functions\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

int Employee::numEmployees = 0;

// default constructor

Employee::Employee()

{

firstName = "Not Given";

lastName = "Not Given";

gender = 'U';

dependents = 0;

annualSalary = 20000;

bens = Benefits();

numEmployees ++;

}

// Employee constructor with multi arguements (remember to use class name as the data type when integrating into parameters)

Employee::Employee(string fn, string ln, char gen, int dep, double sal, Benefits bens\_in )

{

firstName = fn;

lastName = ln;

gender = gen;

dependents = dep;

annualSalary = sal;

bens = bens\_in;// set original instance to parameter instance

numEmployees ++;

}

// create function for calculatePay

double Employee:: calculatePay(double sal){

double wsal = sal/52;

return wsal;

}

// create function to display all employee information

void Employee::displayEmployee()

{

cout<<"\nEmployee Information"<<endl;

cout<<"\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_"<<endl<<endl;

cout<<"Employee Name: \t"<<firstName<<" "<<lastName<<endl;

cout<<"Gender: \t"<<gender<<endl;

cout<<"Dependents: \t"<<dependents<<endl;

cout<<"Annual Salary: \t"<< setprecision(2) << showpoint << fixed << annualSalary<<endl;

cout<<"Weekly Salary: \t"<< setprecision(2) << showpoint << fixed <<calculatePay(annualSalary)<<endl;

bens.displayBenefits();

cout<<endl<<endl;

}

// create getters and setters for all private variables

string Employee::getFirstName(){return firstName;}

void Employee::setFirstName(string fn){firstName=fn;}

string Employee::getLastName(){return lastName;}

void Employee::setLastName(string ln){lastName=ln;}

char Employee::getGender(){return gender;}

void Employee::setGender(char gen){gender=gen;}

int Employee::getDependents(){return dependents;}

void Employee::setDependents(int dep){dependents=dep;}

void Employee::setDependents(string dep){dependents= atoi(dep.c\_str());}

double Employee::getAnnualSalary(){return annualSalary;}

void Employee::setAnnualSalary(double sal){annualSalary=sal;}

void Employee::setAnnualSalary(string sal){annualSalary=atof(sal.c\_str());}

int Employee::getNumEmployees(){return numEmployees;}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Non Class Functions\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// create header function

void DisplayApplicationInformation(){

// create header to introduce the program

cout<<"\n--------------------------------------------"<<endl;

cout<<"| Welcome the Basic User Interface Program |"<<endl;

cout<<"| CIS247, Week 4 Lab |"<<endl;

cout<<"| Name: RENEE THOMAS |"<<endl;

cout<<"| This program updates the employee class |"<<endl;

cout<<"| then outputs the data with benefits |"<<endl;

}

// create divider string function

void DisplayDivider(string outputTitle){

cout<<"\n\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* "<<outputTitle<< " \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"<<endl<<endl;

}

// create function to get information from user

string getInput(string inputType){

// declare variable strInput

string strInput;

// get information from user

cout<<"Enter your " <<inputType;

//put user input into variable strInput

getline(cin, strInput);

//output user input

return strInput;

}

// create good bye message

void TerminateApplication(){

cout<<"\n\nThank you for running the Employee Class program."<<endl;

cout<<"Press any key to leave the program..."<<endl;

}