Nhi Pham

Jeremy Howell

10/12/2021

Project 1

Our Work on the Project:

Nhi Pham worked on setting up the classes and their functions.

Jeremy Howell worked on the tallying of the salaries as well as the output.

main.cpp

```
#include <iostream>
#include "Education.h"
#include "Employee.h"
#include "Faculty.h"
#include "Staff.h"
#include "Partime.h"
#include <vector>
#include <typeinfo>

int main()
{
  float partTimeMonthlyEarning = 0.00;
  float facultyMonthlyEarning = 0.00;
  float staffMonthlyEarning = 0.00;
  float employeeMonthlyEarning = 0.00;

  vector<Employee*> employees;
  employees.push_back(new Staff("Allen", "Paita", "123", 'M', "2/23/59", 50.00));
  employees.push_back(new Staff("Zapata", "Steven", "456", 'F', "7/12/64",
35.00));
  employees.push_back(new Staff("Rios", "Enrique", "789", 'M', "6/2/70", 40.00));
  employees.push_back(new Faculty("Johnson", "Anne", "243", 'F', "4/27/62",
"Full", "Ph.D", "Engineering",3));
  employees.push_back(new Faculty("Bouris", "William", "791", 'F', "3/14/75",
"Associate", "Ph.D", "English", 1));
  employees.push_back(new Faculty("Andrade", "Christopher", "623", 'F',
"5/22/80", "Assistant", "MS", "Physical Education", 0));
  employees.push_back(new Partime("Guzman", "Augusto", "455", 'F', "8/10/77",
35.00, 30));
  employees.push_back(new Partime("Depirro", "Martin", "678", 'F', "9/15/87",
30.00, 15));
```

```
  employees.push_back(new Partime("Aldaco", "Marque", "945", 'M', "11/24/88",
20.00, 35));
  for (int i = 0; i < employees.size(); i++) {
    cout << i+1 << ".";
    employees[i]->putData();
    auto& type = *employees[i];
    string employeeType = typeid(type).name();
    if(employeeType == "7Partime") {
      partTimeMonthlyEarning += employees[i]->monthlyEarning();
    }
    else if(employeeType == "7Faculty") {
      facultyMonthlyEarning += employees[i]->monthlyEarning();
    }
    else if(employeeType == "5Staff") {
      staffMonthlyEarning += employees[i]->monthlyEarning();
    }
    else {
    }
    employeeMonthlyEarning += employees[i]->monthlyEarning();

    cout << endl;
  }

  cout << "Total monthly salary for all the part-time staff: $" <<
partTimeMonthlyEarning << endl;

  cout << "Total monthly salary for faculty: $" << facultyMonthlyEarning << endl;

  cout << "Total monthly salary for all staff: $" << staffMonthlyEarning << endl;

  cout << "Total monthly salary for all employees: $" << employeeMonthlyEarning;

  return 0;
}
```

Employee.h

```
#ifndef EMPLOYEE_H
#define EMPLOYEE_H
#include <iostream>
#include <cstring>
using namespace std;

class Employee {
public:
```

```
  enum Sex {M = 'M', F = 'F'};
  const float FACULTY_MONTHLY_SALARY = 5000.00;
  const int STAFF_MONTHLY_HOURS_WORKED = 160;
  Employee();
  Employee(std::string ln, std::string fn, std::string id, char s, std::string
bd);
  virtual void putData();
  void getLastName();
  void setLastName(string ln);
  void getFirstName();
  void setFirstName(string fn);
  void getID();
  void setID(string id);
  void getGender();
  void setGender(Sex s);
  void getDOB();
  void setDOB(string bd);
  virtual double monthlyEarning() = 0;
private:
  string lastname;
  string firstname;
  string IDnum;
  Sex sex;
  //char birthdate[20] = "N/A";
  string birthdate;
};

#endif
```

Employee.cpp

```
#include "Employee.h"

Employee::Employee() {
  lastname = "N/A";
  firstname = "N/A";
  IDnum = "N/A";
  sex = M;
  birthdate = "N/A";
};

Employee::Employee(std::string ln, std::string fn, std::string id, char s,
std::string bd) {
  lastname = ln;
  firstname = fn;
```

```
  IDnum = id;
  sex = Sex(s);
  string yr = bd.substr(bd.length() - 2, 2);
  birthdate = bd.substr(0, bd.length()-2);
  birthdate += "19" + yr;
};

void Employee::putData(){
  cout << "ID Employee number: " << IDnum << endl;
  cout << "Employee name: " << firstname << " " << lastname << endl;
  cout << "Birth date: " << birthdate << endl;
};

void Employee::getLastName() {
  cout << "Enter last name: " << endl;
  cin >> lastname;
};

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

void Employee::getFirstName() {
  cout << "Enter first name: " << endl;
  cin >> firstname;
};

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

void Employee::getID() {
  cout << "Enter ID: " << endl;
  cin >> IDnum;
};

void Employee::setID(string id) {
  IDnum = id;
};

void Employee::getGender() {
  char gender;
  cout << "Enter gender: " << endl;
  cin >> gender;
  sex = (Sex)gender;
```

```
};

void Employee::setGender(Sex s) {
  sex = s;
};

void Employee::getDOB() {
  cout << "Enter birth date: " << endl;
  cin >> birthdate;
};

void Employee::setDOB(string bd) {
  birthdate = bd;
};
```

Staff.h

```
#ifndef STAFF_H
#define STAFF_H
#include "Employee.h"
class Staff : public Employee {
private:
  double hourlyrate;
public:
  Staff();
  Staff(std::string ln, std::string fn, std::string id, char s, std::string bd,
double hr);
  virtual void getData();
  double getHourlyRate();
  void setData(double hr);
  double monthlyEarning();
  void putData();
};
#endif
```

Staff.cpp

```
#include "Staff.h"
#include "Employee.h"
#include <iostream>
using namespace std;

Staff::Staff():Employee() {
  hourlyrate = 0;
};
```

```
Staff::Staff(std::string ln, std::string fn, std::string id, char s, std::string
bd, double hr):Employee(ln, fn, id, s, bd) {
  hourlyrate = hr;
}

void Staff::getData() {
  Employee::getFirstName();
  Employee::getLastName();
  Employee::getID();
  Employee::getGender();
  Employee::getDOB();
  cout << "Enter hourly rate: " << endl;
  cin >> hourlyrate;
}

double Staff::getHourlyRate() {
  return hourlyrate;
}

void Staff::setData(double hr) {
  hourlyrate = hr;
}

double Staff::monthlyEarning() {
  return hourlyrate * STAFF_MONTHLY_HOURS_WORKED;
}

void Staff::putData() {
  Employee::putData();
  cout << "Full Time" << endl;
  cout << "Monthly Salary: $" << monthlyEarning() << endl;
}
```

Education.h

```
#ifndef EDUCATION_H
#define EDUCATION_H
#include <iostream>
using namespace std;

class Education {
private:
  string degree;
  string major;
```

```
  int research;
public:
  Education();
  Education(string deg, string mj, int re);
  void getDegree();
  void getMajor();
  void getResearch();
  void setDegree(string deg);
  void setMajor(string mj);
  void setResearch(int re);
};
#endif
```

Education.cpp

```
#include "Education.h"
#include <iostream>
#include <string>
using namespace std;

Education::Education() {
  degree = "N/A";
  major = "N/A";
  research = 0;
};
Education::Education(string deg, string mj, int re) {
  degree = deg;
  major = mj;
  research = re;
};
void Education::getDegree() {
  cout << "Enter degree (MS or PhD): " << endl;
  cin >> degree;
};
void Education::getMajor() {
  cout << "Enter major: " << endl;
  getline(cin >> ws, major);
};
void Education::getResearch() {
  cout << "Enter number or researches: " << endl;
  cin >> research;
};
void Education::setDegree(string deg) {
  degree = deg;
};
```

```
void Education::setMajor(string mj) {
  major = mj;
};
void Education::setResearch(int re) {
  research = re;
};
```

Faculty.h

```
#ifndef FACULTY_H
#define FACULTY_H
#include "Employee.h"
#include "Education.h"

class Faculty: public Employee {
public:
  enum Level{AS, AO, FU};
  Faculty();
  Faculty(std::string ln, std::string fn, std::string id, char s, std::string bd,
std::string lv, std::string deg, std::string mj, int re);
  void getData();
  void setLevel();
  double monthlyEarning();
  void putData();
private:
  Level level;
  Education edu;
};
#endif // !FACULTY_H
```

Faculty.cpp

```
#include "Faculty.h"

Faculty::Faculty():Employee(), edu() {
  level = AS;
};
Faculty::Faculty(std::string ln, std::string fn, std::string id, char s,
std::string bd, std::string lv, std::string deg, std::string mj, int re) :
        Employee(ln, fn, id, s, bd) , edu(deg, mj, re) {
  if (lv == "Assistant") {
    level = AS;
  }
  else if (lv == "Associate") {
    level = AO;
```

```
  }
  else {
    level = FU;
  }
};
void Faculty::getData() {
  Employee::getFirstName();
  Employee::getLastName();
  Employee::getID();
  Employee::getGender();
  Employee::getDOB();
  //cout << "Enter level: " << endl; cin >> level;
  //edu.get....()
};
void Faculty::setLevel() {

};
double Faculty::monthlyEarning() {
  switch (level) {
  case AS:
    return Employee::FACULTY_MONTHLY_SALARY;
  case AO:
    return 1.2 * Employee::FACULTY_MONTHLY_SALARY;
  case FU:
    return 1.4 * Employee::FACULTY_MONTHLY_SALARY;
  }
};
void Faculty::putData() {
  Employee::putData();
  switch (level) {
  case AS:
    cout << "Assistant Professor" << endl;
    break;
  case AO:
    cout << "Associate Professor" << endl;
    break;
  case FU:
    cout << "Full Professor" << endl;
    break;
  }
  cout << "Monthly Salary: $" << monthlyEarning() << endl;
};
```

Partime.h

```
#ifndef PARTIME_H
#define PARTIME_H
#include "Staff.h"
class Partime: public Staff {
private:
  int hourperweek;
public:
  Partime();
  Partime(std::string ln, std::string fn, std::string id, char s, std::string bd,
double hr, int hpw);
  void getData();
  void setHourPerWeek(int hpw);
  double monthlyEarning();
  void putData();
};
#endif // !PARTIME_H
```

Partime.cpp

```
#include "Partime.h"

Partime::Partime():Staff() {
  hourperweek = 0;
}
Partime::Partime(std::string ln, std::string fn, std::string id, char s,
std::string bd, double hr, int hpw) : Staff(ln, fn, id, s, bd, hr) {
  hourperweek = hpw;
};
void Partime::getData() {
  Staff::getData();
  cout << "Enter hours worked per week: " << endl;
  cin >> hourperweek;
};
void Partime::setHourPerWeek(int hpw) {
  hourperweek = hpw;
};
double Partime::monthlyEarning() {
  return Staff::getHourlyRate() * hourperweek * 4;
};
void Partime::putData() {
  Employee::putData();
  cout << "Hours works per month: " << hourperweek * 4 << endl;
  cout << "Monthly Salary: $" << monthlyEarning() << endl;
};
```

Output:

```
./main
1.ID Employee number: 123
Employee name: Paita Allen
Birth date: 2/23/1959
Full Time
Monthly Salary: $8000

2.ID Employee number: 456
Employee name: Steven Zapata
Birth date: 7/12/1964
Full Time
Monthly Salary: $5600

3.ID Employee number: 789
Employee name: Enrique Rios
Birth date: 6/2/1970
Full Time
Monthly Salary: $6400

4.ID Employee number: 243
Employee name: Anne Johnson
Birth date: 4/27/1962
Full Professor
Monthly Salary: $7000

5.ID Employee number: 791
Employee name: William Bouris
Birth date: 3/14/1975
Associate Professor
Monthly Salary: $6000

6.ID Employee number: 623
Employee name: Christopher Andrade
Birth date: 5/22/1980
Assistant Professor
Monthly Salary: $5000

7.ID Employee number: 455
Employee name: Augusto Guzman
Birth date: 8/10/1977
Hours works per month: 120
Monthly Salary: $4200
```

```
8.ID Employee number: 678
Employee name: Martin Depirro
Birth date: 9/15/1987
Hours works per month: 60
Monthly Salary: $1800

9.ID Employee number: 945
Employee name: Marque Aldaco
Birth date: 11/24/1988
Hours works per month: 140
Monthly Salary: $2800

Total monthly salary for all the part-time staff: $8800
Total monthly salary for faculty: $18000
Total monthly salary for all staff: $20000
Total monthly salary for all employees: $46800
```

Demonstrated at 11:04 am on 10/12/2021