Sheharyar Khan
CS-116

LAB 2
Employee Class

Source Code

header.h

```
#ifndef EMPLOYEE_H
#define EMPLOYEE_H

#include <iostream>
#include <string>

using namespace std;

class Employee
{
   private:    //member variables
   string name;
   int idNumber;
   string department;
   string position;

   public: //mutator and accessor functions
   Employee(string eName, int eIdNumber, string eDepartment, string ePostion);
   Employee(string eName, int eIdNumber);
   Employee();

   string getName();
   int getIdNumber();
   string getDepartment();
   string getPosition();

   void setName(string eName);
   void setIdNumber(int eIdNumber);
   void setDepartment(string eDepartment);
   void setPosition(string ePosition);

};

Employee::Employee(string eName, int eIdNumber, string eDepartment, string ePosition)
{
   name = eName;
   idNumber = eIdNumber;
   department = eDepartment;
   position = ePosition;
}

Employee::Employee(string eName, int eIdNumber)
{
   name = eName;
   idNumber = eIdNumber;
   department = "";
   position = "";
}

Employee::Employee()
{
   name = "";
   idNumber = 0;
   department = "";
   position = "";
}

string Employee::getName()
{
   return name;
}
int Employee::getIdNumber()
```

```
{
   return idNumber;
}
string Employee::getDepartment()
{
   return department;
}
string Employee::getPosition()
{
   return position;
}

void Employee::setName(string eName)
{
   name = eName; //simply sets the attribute to the input in the parameter
}
void Employee::setIdNumber(int eIdNumber)
{
   idNumber = eIdNumber;
}
void Employee::setDepartment(string eDepartment)
{
   department = eDepartment;
}
void Employee::setPosition(string ePosition)
{
   position = ePosition;
}

void displayEmployee(Employee employee)
/*this function is not part of the class because of the way it is called in the driver
function. The way its called suggests that it is function outside the class scope.*/
{
   cout<<"Name: "<<employee.getName()<<endl;
   cout<<"ID Number: "<<employee.getIdNumber()<<endl;
   cout<<"Department: "<<employee.getDepartment()<<endl;
   cout<<"Position: "<<employee.getPosition()<<endl<<endl;
}


#endif
```

printmefirst.h

```
#ifndef PRINTMEFIRST_H
#define PRINTMEFIRST_H

#include <iostream>
#include <ctime>
#include <iomanip>

#include "header.hpp"

using namespace std;

void printMeFirst(string name, string courseInfo)
{
cout <<" Program written by: "<< name << endl; // put your name here
cout <<" Course info: "<< courseInfo << endl;
time_t now = time(0); // current date/time based on current system
char* dt = ctime(&now); // convert now to string for
cout << " Date: " << dt << endl;
}

#endif
```

EmployeeClass.cpp

```
/**
*TODO:The purpose of this program is to keep track of employee information such as name, ID
number, Departmant, and position
@param name - Sheharyar Khan
@param courseInfo - CS-116
@return - none
*/

#include <iostream>
#include "header.hpp" //using a header file for the class and the functions
#include "printmefirst.hpp" //using a separate header file for the print me first function as
it will probably be used in every project for this course and this would be easier than
copying the code over and over again. I also wanted to apply what I learned today in class

using namespace std;

int main()
{
printMeFirst ("Sheharyar Khan", "CS - 116: Lab 2"); //put your name instead of "Ron Sha"
// Create an Employee object to test constructor #1.
Employee susan ("Susan Meyers", 47899, "Accounting", "Vice President");
// Create an Employee object to test constructor #2.
Employee mark ("Mark Jones", 39119);
mark.setDepartment("IT");
mark.setPosition("Programmer");
// Create an Employee object to test constructor #3.
Employee joy;
joy.setName("Joy Rogers");
joy.setIdNumber(81774);
joy.setDepartment("Manufacturing");
joy.setPosition("Engineer");

// Display each employee's data.
displayEmployee(susan);
displayEmployee(mark);
displayEmployee(susan);
return 0;
}
```

Purpose

To make a class that works with the driver function. The class had to have member variables, mutators and accessors.

Logic

In this class I am using the data types requested in the lab requirements. I made the functions as efficient as possible. The variables are private because we don't want them to interact with the user directly. Member functions are public because we want the user to be able to use them to access the variables. The mutator functions take 1 parameter and set the object's attribute to it respectively for example:

```
void setName(string eName);
```

When this function is called it sets *eName* equal to the *name* variable in the object class. Similarly when *setDepartment* or *setIdNumber* are called they set the objects variables equal to the variable provided.

I am using a header file for the print me first function because it will be required in probably almost every lab so I thought it would be perfect to implement what I learned in class and it worked out great. It saves me from looking for the function definition everytime and saves time. I also used a header for the class because I wanted to practice using header files. I learned that I need to include the <*iostream*> library if I want to use cout and I must also specify that I am *using namespace std;* .

Test Case: