Amir Sotoodeh
CS256
5/22/18

Project: Exceptions Handling
Github:
https://github.com/excisionhd/CS256/blob/master/ExceptionsProject/ExceptionsProject.cpp

```
/*****************************************************
* FILENAME : ExceptionProject.cpp
*
* DESCRIPTION :
*       Project dealing with Exception handling.
*
* AUTHOR :  Amir Sotoodeh
* START DATE : 5/22/18
*
*****************************************************/

#include "stdafx.h"
#include <string>
#include <iostream>

using namespace std;

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

public:
	class InvalidEmployeeNumber {
	private:
		int value;
	public:
		InvalidEmployeeNumber(int v) {
			value = v;
		}
		int getValue() {
			return value;
		}
	};
	Employee(string nm, int id, string dept, string pos) {
		name = nm;
		idNumber = id;
		department = dept;
```

```
		position = pos;
	}
	Employee(string nm, int id) {
		name = nm;
		idNumber = id;
		department = "";
		position = "";
	}
	Employee() {
		name = "";
		idNumber = 0;
		department = "";
		position = "";
	}

	void setName(string nm) {
		name = nm;
	}
	void setIDNumber(int id) {
		if (id < 0 || id > 9999) {
			throw InvalidEmployeeNumber(id);
		}
		else
			idNumber = id;
	}
	void setDepartment(string dept) {
		department = dept;
	}
	void setPosition(string pos) {
		position = pos;
	}

	string getName() {
		return name;
	}
	int getIDNumber() {
		return idNumber;
	}
	string getDepartment() {
		return department;
	}
	string getPosition() {
		return position;
	}
};

class ProductionWorker : public Employee {
private:
```

```
	int shift;
	double hourlyRate;
public:
	class InvalidShift {
	private:
		int value;
	public:
		InvalidShift(int v) {
			value = v;
		}
		int getValue() {
			return value;
		}
	};
	class InvalidPayRate {
	private:
		int value;
	public:
		InvalidPayRate(int v) {
			value = v;
		}
		int getValue() {
			return value;
		}
	};
	ProductionWorker() {
	}
	ProductionWorker(int s, double h) {
		shift = s;
		hourlyRate = h;
	}

	void setShift(int s) {
		if (s < 0 || s > 1) {
			throw InvalidShift(s);
		}
		else
			shift = s;
	}
	void setHourlyPayRate(int h) {
		if (h<0) {
			throw InvalidPayRate(h);
		}
		else
			hourlyRate = h;
	}
};
```

```
int main() {
	Employee Susan("Susan Meyers", 47899, "Accounting", "Vice President");
	Employee Mark("Mark Jones", 39119);
	Employee Joy;

	Mark.setDepartment("IT");
	Mark.setPosition("Programmer");

	Joy.setName("Joy Rogers");
	Joy.setIDNumber(999);
	Joy.setDepartment("Manufacturing");
	Joy.setPosition("Engineer");

	cout << "Name\t\t" << "ID Number\t" << "Department\t" << "Position" <<
endl;
	cout << Susan.getName() << "\t" << Susan.getIDNumber() << "\t\t" <<
Susan.getDepartment() << "\t" << Susan.getPosition() << endl;
	cout << Mark.getName() << "\t" << Mark.getIDNumber() << "\t\t" <<
Mark.getDepartment() << "\t\t" << Mark.getPosition() << endl;
	cout << Joy.getName() << "\t" << Joy.getIDNumber() << "\t\t" <<
Joy.getDepartment() << "\t" << Joy.getPosition() << endl << endl;

	//Will catch error!
	try {
		Joy.setIDNumber(99999);
	}
	catch (Employee::InvalidEmployeeNumber e) {
		cout << "Please enter a valid employee number (0-9999)!" << endl;
		cout << to_string(e.getValue()) << " is not a valid entry!" <<
endl << endl;
	}

	//Will not catch error, proper value entered.
	try {
		Joy.setIDNumber(2);
	}
	catch (Employee::InvalidEmployeeNumber e) {
		cout << "Please enter a valid employee number (0-9999)!" << endl;
		cout << to_string(e.getValue()) << " is not a valid entry!" <<
endl;
	}

	ProductionWorker Bob(1,0);

	//Will catch error!
	try {
		Bob.setShift(2);
	}
```

```
	catch (ProductionWorker::InvalidShift e) {
		cout << "Please enter a valid shift (1 or 0)!" << endl;
		cout << to_string(e.getValue()) << " is not a valid entry!" <<
endl << endl;
	}

	//Will not catch error, proper value entered.
	try {
		Bob.setShift(1);
	}
	catch (ProductionWorker::InvalidShift e) {
		cout << "Please enter a valid shift (1 or 0)!" << endl;
		cout << to_string(e.getValue()) << " is not a valid entry!" <<
endl;
	}

	//Will catch error!
	try {
		Bob.setHourlyPayRate(-1);
	}
	catch (ProductionWorker::InvalidPayRate e) {
		cout << "Please enter a valid pay rate!" << endl;
		cout << to_string(e.getValue()) << " is not a valid entry!" <<
endl << endl;
	}

	//Will not catch error, proper value entered.
	try {
		Bob.setHourlyPayRate(1);
	}
	catch (ProductionWorker::InvalidPayRate e) {
		cout << "Please enter a valid pay rate!" << endl;
		cout << to_string(e.getValue()) << " is not a valid entry!" <<
endl;
	}

	return 0;
}
```

Output

```
Name            ID Number       Department      Position
Susan Meyers    47899           Accounting      Vice President
Mark Jones      39119           IT              Programmer
Joy Rogers      999             Manufacturing   Engineer

Please enter a valid employee number (0-9999)!
99999 is not a valid entry!

Please enter a valid shift (1 or 0)!
2 is not a valid entry!

Please enter a valid pay rate!
-1 is not a valid entry!

Press any key to continue . . .
```