

Object-Oriented Programming Language

10/11/2018

Homework Assignment No. 4

Due 11:59 pm, **Thursday October 18, 2018**

Late submission within 24 hours: score*0.9;

Late submission within one week: score*0.8.

The solutions will be posted after one week of the due date.

(Total 150%)

1. **(50%):** Please complete four functions. Two of these functions are part of the selection sort algorithm, which is to sort a list of numbers into increasing order. (You can google how selection sort works.)

The first function is to find, from the `start` position, the min value from a vector `vec` of integers. Please implement this function with your own logic. In other words, please do NOT use the utilities provided by `c++` for the min value seeking. The return value is the position of the min value in the vector. The function declaration is as follows:

```
| int findMin(const vector<int>& vec, int start);
```

The second function is to switch two integers based on their positions in the vector `vec`. The function declaration is as follows:

```
| void switchPos(vector<int>& vec, int pos1, int pos2);
```

The third function is the overall logic of the selection sort. You will use the first two functions in this function to complete the selection sort algorithm.

```
| void selectionSort(vector<int>& vec);
```

The forth function is to display the vector.

```
| void printOut(string s, const vector<int>& vec);
```

The following is the main function you cannot change.

```
| int main(){  
|     vector<int> v;  
|     int t;  
|     cout << "Please input the numbers you wish to sort, and stop  
| with any alphabet: ";  
|     while(cin >> t) v.push_back(t);  
|     printOut("Original", v);  
|     selectionSort(v);  
|     printOut("Sorted", v);  
|     return 0;
```

| }

Output examples:

```
Please input the numbers you wish to sort, and stop with any alphabet: 5 4 3 2 1 b
Original: 5 4 3 2 1
Sorted: 1 2 3 4 5
```

```
Please input the numbers you wish to sort, and stop with any alphabet: 1 4 3 8 9 1 3 2 5 4 3 9 7 8 6 3 q
Original: 1 4 3 8 9 1 3 2 5 4 3 9 7 8 6 3
Sorted: 1 1 2 3 3 3 3 4 4 5 6 7 8 8 9 9
```

2. (50%): Please extend Problem 1 in Lab 4. You will need to add a boolean data member called `gender`, a no-argument constructor for the `Person` class in which you will randomly generate a person's age and gender. The name of a female person will be Jane, and that for a male person is John.

For the `PersonVector` class, you will implement an overloading function for the `addPerson(int number)` function by giving the argument of how many people the user wishes to generate randomly, and it will use the no-argument constructor of the `Person` class for the generation of the persons.

The no-argument constructor starts with the following code fragment:

```
Person::Person() {
    static bool seeded = false;
    if(!seeded){
        srand(time(0));
        seeded = true;
    }
    ...
}
```

The following is the main method you cannot change:

```
#include <iostream>
#include "PersonVector.h"

int main(){
    int n;
    std::cout << "How many people to generate? ";
    std::cin >> n;
    PersonVector pv;
    pv.addPersons(n);
    pv.displayPersons();
}
```

The following are sample runs of the program:

```
How many people to generate? 3
Name    Age    Gender
Jane    47     0
Jane    90     0
Jane    30     0
```

```
How many people to generate? 5
Name    Age    Gender
John    94      1
Jane    17      0
Jane    99      0
John    62      1
Jane    17      0
```

```
How many people to generate? 0
Name    Age    Gender
```

3. (50%): Please extend Problem 2 in Lab 4. You will provide the reversed conversion from Rectangle to Square, if it is possible.

The following is the main method you cannot change:

```
int main () {
    cout << "Please input the side of a square: ";
    int side;
    cin >> side;

    Rectangle rect;
    Square sqr(side);
    rect.convert(sqr);

    cout << "The Rectangle has:" << endl;
    rect.print();

    cout << "Please input the width and height of a rectangle: ";
    int width, height;
    cin >> width >> height;

    Rectangle rect2(width, height);
    Square sqr2;
    sqr2.convert(rect2);

    cout << "The square has:" << endl;
    sqr2.print();

    return 0;
}
```

The following are sample runs of the program:

```
Please input the side of a square: 3
The Rectangle has:
width: 3, height: 3, and area: 9
Please input the width and height of a rectangle: 4 4
Successful conversion to Square.
The square has:
side: 4, and area: 16
```

Object-Oriented Programming Language

10/11/2018

```
Please input the side of a square: 3
The Rectangle has:
width: 3, height: 3, and area: 9
Please input the width and height of a rectangle: 3 4
Conversion Error! Square not changed.
The square has:
side: 1, and area: 1
```