Part E - Polymorphism

Function Templates

Workshop 10 (out of 10 marks - 3% of your final grade)
[Ver. 1.2]

In this workshop, you will code a function template for validating a value between a minimum and maximum range.

LEARNING OUTCOMES

Upon successful completion of this workshop, you will have demonstrated your abilities to

- code a function template
- reflect on what you have learned from this workshop
- reflect on this semester overall

SUBMISSION POLICY

The "in-lab" section is to be completed during your assigned lab section. It is to be completed and submitted by the end of the workshop period. If you attend the lab period and cannot complete the in-lab portion of the workshop during that period, ask your instructor for permission to complete the in-lab portion after the period. If you do not attend the workshop, you can submit the “in-lab” section along with your “at-home” section (with a penalty; see below). The “at-home” portion of the lab is due on the day of your next scheduled workshop (23:59).

All your work (all the files you create or modify) must contain your name, Seneca email and student number.

You are responsible to regularly back up your work.

Late submission penalties:

- In-lab submitted late, with at-home: Maximum of 20/50 for in-lab and Maximum of 50/50 for at home
- Workshop late for one week: in-lab, at-home and reflection must all be submitted for maximum of 50 / 100

- Workshop late for more than one week: in-lab, at-home and reflection must all be submitted for maximum of 30 / 100
- If any of in-lab, at-home or reflection is missing the mark will be zero.

IN LAB (40%) – VALIDATE FUNCTION TEMPLATE

Create a validate function template in a header file named **`validate.h`**. that accepts five arguments (a minimum, a maximum, an array of test-values, number of elements in the array and an array of bools for validation results) and returns a bool that is true only if all the test-value elements are between the minimum and maximum arguments, and false otherwise, that is:

Return true if for all the elements of the array, testValue[i] >= minimum && testValue[i] <= maximum and return false otherwise (having "i" being the index of all the elements in the testValue array)

Also as the validation test is being done on each element, the result of validation is stored in the corresponding element of the bool array so the caller program can determine exactly which of elements of the testValue array are invalid.

Make sure the function is designed in a way that all test objects (min, max and test elements) are not copied, or modified in the function.

Client Module

The main program that uses your implementation is listed below:

```
// OOP244 Workshop 10 - Templates
// File: w10.cpp
// Version: 1.0
// Date: 2017/04/09
// Author: Fardad Soleimanloo
// Description:
// This file tests the in-lab section of your workshop
////////////////////////////////////////////


#include <iostream>
#include "Car.h"
#include "Employee.h"
#include "Student.h"
#include "validate.h"
using namespace std;
using namespace ict;
int main(){
  int i;
```

```
  bool v[10];
  Student S[6] = { { 2345, "Lisa Simpson" }, { 12345, "Bart Simpson" },
                   { 4567, "Ralph Wiggum" }, { 56789, "Milhouse Van Houten" },
                   { 67890, "Todd Flanders" }, { 34567, "Nelson Muntz" } };
  Employee E[6] = { Employee(213456, "Carl Carlson", 62344.56), Employee(122345, "Mindy
Simmons", 65432.44),
                    Employee(435678, "Lenny Leonard", 43213.22), Employee(546789, "Waylon
Smithers", 654321.55),
                    Employee(657890, "Frank Grimes", 34567.88), Employee(364567, "Homer
Simpson", 55432.11) };
  int vals[10] = { 2, 6, 4, 67, 4, 6, 7, 5, 4, 6 };
  char cvals[10] = { 'A','e','B', 'd', 'e', 'G', 'H', 'l', 'p','x' };
  Car C[5] = { Car("DEFGHI", "Tesla Model S"), Car("BBCDEF", "BMW 320"),
               Car("CDEFG", "Ford Festiva"), Car("BCDEFG", "Ford Festiva"),
               Car("AFGHIJ", "Nissan Maxima") };
  if (!validate(Student(11111), Student(99999), S, 6, v)){
    cout << "These students have invalid student ids:" << endl;
    for (i = 0; i < 5; i++){
      if (!v[i]) cout << S[i] << endl;
    }
  }
  else{
    cout << "All students have valid students ids" << endl;
  }
  if (!validate(Employee(111111), Employee(999999), E, 6, v)){
    cout << "These employee have invalid employee ids:" << endl;
    for (i = 0; i < 5; i++){
      if (!v[i]) cout << S[i] << endl;
    }
  }
  else{
    cout << "All employee have valid employee ids" << endl;
  }
  if (!validate(3, 7, vals, 10, v)){
    cout << "These integer values are invalid: " << endl;
    for (i = 0; i < 10; i++){
      if (!v[i]) cout << vals[i] << endl;
    }
  }
  else{
    cout << "All integer values are valid" << endl;
  }
  if (!validate(Car("BBBBBB"), Car("ZZZZZZ"), C, 5, v)){
    cout << "These cars have invalid license plates:" << endl;
    for (i = 0; i < 5; i++){
      if (!v[i]) cout << C[i] << endl;
    }
  }
  else{
    cout << "All cars have valid license plates" << endl;
  }
  if (!validate('B', 'P', cvals, 10, v)){
    cout << "These character values are invalid: " << endl;
    for (i = 0; i < 10; i++){
      if (!v[i]) cout << cvals[i] << endl;
    }
  }
  else{
```

```
    cout << "All character values are valid" << endl;
  }
  return 0;
}
```

Output:

```
These students have invalid student ids:
2345 Lisa Simpson
4567 Ralph Wiggum
All employee have valid employee ids
These integer values are invalid:
2
67
These cars have invalid license plates:
AFGHIJ Nissan Maxima
These character values are invalid:
A
e
d
e
l
p
x
```

IN-LAB SUBMISSION

To test and demonstrate execution of your program use the same data as the output example above.

If not on matrix already, upload **all the .h and .cpp files** and **244_w10_tester.cpp** to your matrix account. Compile and run your code and make sure everything works properly.

Then run the following script from your account:

~profname.proflastname/submit 244_w10_lab <ENTER>

and follow the instructions.

Please note that a successful submission does not guarantee full credit for this workshop.

If the professor is not satisfied with your implementation, your professor may ask you to resubmit. Resubmissions will attract a penalty.

AT-HOME (60%)

Please provide answers to the following questions in a text file named reflect.txt**.**

WORKSHOP QUESTIONS:

1- Name all different types of polymorphism with an example
2- What category of polymorphism does a template fall?
3- What happens if a function template is never called?

SUBJECT SURVEY:

4- What was the most interesting thing you learned this semester?
5- Do you feel that the quizzes about the week's readings helped you learn more than you might have otherwise done.
6- Are there any things that you particularly like about the way that the course is delivered?
7- Are there any things that you particularly dislike about the way that the course is delivered?
8- Is there anything you would like to see added to the way the course is delivered?
9- How would you rate your level of understanding of the course topics?
    a. Very good
    b. Pretty good
    c. Adequate
    d. Poor
10- Did you enjoy doing the workshops? Why?
11- The content of this course was
    a. Too little
    b. Just right
    c. Too much

AT-HOME SUBMISSION

If not on matrix already, upload **reflect.txt** to your matrix account.

Then run the following script from your account: (replace profname.proflastname with your professors Seneca userid)

```
~profname.proflastname/submit 244_w10_home <ENTER>
```

and follow the instructions.

Please note that a successful submission does not guarantee full credit for this workshop.

If the professor is not satisfied with your implementation, your professor may ask you to resubmit. Resubmissions will attract a penalty.