

Function Templates

Workshop 10

In this workshop, you code a template for converting a mark to a letter grade.

LEARNING OUTCOMES

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

- code a function template
- code an explicit specialization for a function template
- reflect on what you have learned from this workshop

PART A: GRADEIT - BASIC TASK

Definition

Create a function that takes in two integers (a mark and a maximum possible mark) and returns a single character for the corresponding grade, using the following mapping (where i is the percentage corresponding to the mark):

Percentage	Grade
$i > 79$	A
$69 < i \leq 79$	B
$59 < i \leq 69$	C
$50 < i \leq 59$	D
$i \leq 50$	F

The prototype for the function should be as follows

```
char gradeIt(int mark, int maxMark);
```

Template

Store a template for your function in a header file named `grade.h`.

Explicit Specialization

Code an explicit specialization of your template for `char` inputs, where the maximum mark is 'A', and which returns the mark.

Add your specialization to the header file named `grade.h`.

Client Module

The main program that uses your implementation is listed on the left, while the results are shown on the right:

```
// Workshop 10 - gradeIt
// w10.cpp

#include <iostream>
#include "grade.h"

int main() {
    int t1Marks[] = { 28, 44, 12, 30 };
    int t1MaxMark = 47;
    double t2Marks[] = { 28.0, 22.5, 24.8, 21.6 };
    double t2MaxMark = 47.0;
    char t3Marks[] = { 'A', 'D', 'C', 'F' };
    char t3MaxMark = 'A';

    for (int i = 0; i < 4; i++)
    {
        std::cout << gradeIt(t1Marks[i], t1MaxMark) << " ";
        std::cout << gradeIt(t2Marks[i], t2MaxMark) << " ";
        std::cout << gradeIt(t3Marks[i], t3MaxMark) << std::endl;
    }
    std::cout << std::endl;
}
```

```
D C A
A F D
F D C
C F F
```

Typing script

Create a script of your complete solution using the following commands:

```
+ At the prompt, type: script w10.txt
+ At the prompt, type: whoami
+ At the prompt, type: cat grade.h
+ At the prompt, type: g++ -o w10 w10.cpp
+ At the prompt, type: w10
+ At the prompt type: exit
```

These commands will produce a file named `w10.txt`. Download this file to your local computer.

PART 2: GRADEIT WITH A CLASS

Upgrade your `gradeIt` template to take in two different types and use safe casting to cast the second type to the first type. Provide a specialization that takes in a `Student` as the first type and `int` as the second type (paste the `Student` class definition below into your `grade.h`) and calculates the grade using the `Student`'s `mark` member variable.

Client Module

The main program that uses your implementation is listed on the left, while the results are shown on the right:

```
////////////////////////////////////
//(To be pasted into your grade.h)
class Student{
    double mark;
public:
    Student(double i){ mark = i; };
    double getMark(){ return mark; };
};
////////////////////////////////////

// Workshop 10 - GradeIt with a class
// w10B.cpp

#include <iostream>
#include "grade.h"

int main() {
    int maxMark = 100;
    double mark = 69.55;
    int mark2 = 69;
    Student fred(mark);
    std::cout << gradeIt(mark, maxMark) << std::endl;
```

```
std::cout << gradeIt(mark2, maxMark) << std::endl;  
std::cout << gradeIt(fred, maxMark) << std::endl;  
}
```

B
C
B

Typing script

Create a script of your complete solution using the following commands:

```
+ At the prompt, type: script w10B.txt  
+ At the prompt, type: whoami  
+ At the prompt, type: cat grade.h  
+ At the prompt, type: g++ -o w10b w10B.cpp  
+ At the prompt, type: w10b  
+ At the prompt type: exit
```

These commands will produce a file named **w10B.txt**. Download this file to your local computer.

SUBMISSION

Submit your w10.txt and w10B.txt files through the following email.

nargis.khan@senecacollege.ca