# 34.1 LAB: Passing score count (range-based for loop with auto)

Complete function `CountScore()` that takes in a vector containing the student information of a course and a passing score as parameters. `CountScore()` returns the number of students having a final score that is higher or equal to a given passing score.

The main program parses the student information of a course from a text file into a vector. Information of each student is stored as a `StudentInIntroToComputerScienceCourse` object containing the following data members:

- `name` (string)
- `semester` (string)
- `finalScore` (double)

Use a range-based for loop with `auto` to simplify the iterations of a vector. When the name of the object type of the elements in a vector is long and difficult to be remembered, `auto` is useful in handling the object type without running the risk of typing the wrong name in the code.

Ex: If the input of the program (the passing score) is:

```
65.5
```

`CountScore()` returns 3, and the program output is:

```
Number of students passed: 3
```

Note: main() parses the student information from *StudentInfo.txt*. The auto-grader will use files containing different sets of student information for testing.

539740.3879454.qx3zqy7

**LAB ACTIVITY** 34.1.1: LAB: Passing score count (range-based for loop with auto) 0 / 10

Downloadable files

`StudentInfo.txt` **Download**

Current file: **main.cpp** ▾ **Load default template...**

```
#include <string>
#include <iostream>
#include <iomanip>
```

```
#include <fstream>
#include <vector>
#include "StudentInIntroToComputerScienceCourse.h"

using namespace std;

// FillRoster() parses student information into a vector of
// StudentInIntroToComputerScienceCourse objects
void FillRoster(string studentFile, vector<StudentInIntroToComputerScienceCo
   string studentName;
   string studentSemester;
   double studentScore;
```

**Develop mode** **Submit mode**

Run your program as often as you'd like, before submitting for grading. Below, type any needed input values in the first box, then click **Run program** and observe the program's output in the second box.

Enter program input (optional)

If your code requires input values, provide them here.

**Run program**

Input (from above) → **main.cpp** (Your program) → Outp

Program output displayed here

Coding trail of your work [What is this?]

```
History of your effort will appear here once you begin working
on this zyLab.
```