Custom Exception Class Solutions

Custom Exception Class

- When we write our own exception class, why is it better to derive from a subclass of std::exception hierarchy than to write a completely new class?
  - If we derive from std::exception, we can use its interface instead of having to design our own
  - We can inherit an existing implementation instead of having to write our own
  - Our exception will automatically be handled by existing std::exception catch blocks, without having to add any code
- Why do we not derive directly from std::exception?
  - It only has a default constructor
  - No provision for passing a custom error message

Custom Exception Class Requirements

- When we write our own exception class, which member functions should we consider implementing?
  - Constructor which takes a const std::string
  - Constructor which takes a const char *
  - Copy constructor
  - Override of what() virtual function

invalid_student_grade Members

- In the invalid_student_grade class, why did we not provide any data members?
  - The only data needed is the error string, which is a member of std::out_of_range
- Why did we not implement a copy constructor or destructor for the class?
  - There are no data members, so we do not need to do anything special when copying or destroying objects of this class
  - The compiler generated defaults will be acceptable
  - Or we can use =default

invalid_student_grade Example

- Implement a suitable invalid_student_grade exception class
- Write a program which reads a number from standard input and stores it in a StudentGrade object
- If an invalid grade is entered, an invalid_student_grade exception is thrown
- Your program should handle any exceptions which are thrown during this process
- Test your program, entering both valid and invalid grades
- (The source code for the StudentGrade classes is provided on the next page)

StudentGrade Class Definition

```
class StudentGrade {
   int grade;
  public:
   StudentGrade(int grade) : grade(grade) {
      if (grade < 0)
         throw invalid_student_grade("Invalid grade");
      if (grade > 100)
         throw invalid_student_grade();
   }
};
```