Homework 13: Exceptions

Instructor: Mehmet Emre

CS 32 Spring '22

Due: 5/25 12:30pm
Name & Perm #: Bharat Kathi (5938444)

Homework buddy (leave blank if you worked alone):
Reading: PS 16.1, 16.2

1

1. (4 pts) C++ has an mechanism for throwing exceptions. Many other languages have this feature as well. One interesting way that exception handing varies from one language to another is what type of "thing" can be thrown. Java, for example, has a class called Exception (specifically, `java.lang.Exception`). The only things you can "throw" in Java are objects of type Exception (or of types that are derived via inheritance from Exception). What is the case for C++ - i.e., what kind of value can be thrown in C++?

   Answer: In C++, any primitive type of object can be thrown as an exception.

2. (4 pts) In both C++ and Java, programmers often create special classes for programmer-defined Exceptions. For example, for a project that is doing analysis of word occurrences on Subreddits of the site reddit.com, you might create a NoSuchSubreddit exception for the case where the program is trying to access a Subreddit that does not exist.

   In Java, if you want to look at the code and find out whether a particular class is being used as an exception, it is easy: you see whether it inherits from java.lang.Exception either directly or indirectly. In C++ what do you look for in the code to indicate whether a particular class is used as an exception?

   Answer: An exception in C++ is just a class. It can be recognized by the way that it is actually being used. For example, if the class is used in a throw statement. You can also look for the C++ STL class std::exception. Not every exception class inherits from this in C++ however, so this may not always be the right approach.

2

The author of PS makes the point that *exception classes can be trivial*. We also had some trivial exception classes in the lecture.

1. (2 pts) Illustrate this point by declaring a complete class specification for a C++ `NoSuchStudent` exception. You should have no difficulty confining your answer to the tiny space given here:

   Answer: .

   ```
   class NoSuchStudent{};
   ```

2. (4 pts) Slightly more subtle is this: if the class specification is so trivial, what good it is? Why even declare such a class at all? Be very specific and precise in your answer, but keep it short. (Note: This one actually may require some thought - the answer is not just "lying there" in the book waiting to be found. You will have to really read, digest, and think about the material a bit. Learning may take place. Don't worry - this is a perfectly normal reaction, and any pain you experience will subside.)

   Answer: Even if our exception class is trivial, it will still help the code to know which catch block to execute. This is useful if you have multiple exceptions that can be thrown in a single function.

3

A standard "Hello World" type example for exception handling is *divide by zero*.

1. (2 pts) Write C++ code that creates a (trivial) class for a `DivideByZero` exception.

   Answer: .

   ```
   class DivideByZero{};
   ```

2. (8 pts) Using that class, write the a C++ function named `rationalAsDouble` that takes two arguments: **int** numerator, and **int** denominator and returns a **double**. It coverts these two integers as a fraction into a double. The function should throw a `DivideByZero` exception if the denominator is zero.

   Answer: .

   ```
   double rationalAsDouble(int numberator, int denominator) {
       try {
           if (denominator == 0) throw DivideByZero();
           return (double) numberator / (double) denominator;
       } catch (DivideByZero) {
           cout << "Cannot divide by zero!";
       }
   }
   ```