CS 6311 – Programming Languages I – Fall 2006
Quiz #10 (20 pts)
Monday, 11/20/6

NAME: KEY

1. True or False. In C#, a caught exception can be explicitly rethrown. (2 pts)

   True.

2. Given SystemException is a base class of InvalidCastException, when would the SystemException catch block be executed, given the method called ProcessData, which has the following format: (3 pts)

```
public void ProcessData() {
  try {
    // some statements
  }
  catch(InvalidCastException) {
    // some statements
  }
  catch(SystemException) {
    // some statements
  }
}
```

   a. The code would not compile because the catch blocks are in the wrong order.
   b. Immediately after a SystemException is thrown that is not an InvalidCastException. // Correct answer
   c. Immediately after an InvalidCastException is thrown after executing the InvalidCastException handler.
   d. After the try block exits, but only if an SystemException was thrown.
   e. After the try block exits, and only if a SystemException was thrown that is not an InvalidCastException.

3. Explain what happens if a C# program does not include an exception handler for a thrown exception. (5 pts)

   The exception is propagated up the call stack until it reaches the main method, at which point the exception is thrown to the virtual machine (CLR) and the program terminates. When the exception is thrown to the CLR it prints out information regarding the exception and the call stack when the exception occurred.

4. Write a C# public method called Divider. The method takes two parameters: an integer array and an integer value. The method should modify the array by dividing every element in the array by the passed in integer value. In addition, the code should handle the following exceptions that could occur, as follows: (10 pts)

    a. DivideByZeroException: The exception handler needs to display to the screen, A divide by zero error occurred.
    b. IndexOutOfRangeException: The exception handler should print out the simple message associated with the created exception object.
    c. NullReferenceException: The class of exceptions should be thrown to the calling method.

    Note: The DivideByZeroException and IndexOutOfRangeException classes are on the same hierarchical level.

```
public void Divider(int[] array, int value)
{
        try
        {
                for (int i=0; i<array.Length; i++)
                        array[i] = array[i] / value;
        }
        catch (DivideByZeroException)
        {
                Console.WriteLine("A divide by zero error occurred.");
        }
        catch (IndexOutOfRangeException ex)
        {
                Console.WriteLine(ex.Message);
        }

        // Note: since the NullReferenceException doesn't have a handler it will
        // automatically be propagated up to the calling method.

}
```