York University **Quiz 01 / Fall 2008**
Department of Computer Science and Engineering **CSE 3403**

- The **weight of this test is 5% of the total course mark**.
- This is a closed book, **40 minutes** test.
- **No questions are allowed during the test. If in doubt, write down your doubts and assumptions and proceed with your answer**

LAST NAME	SOLUTIONS
FIRST NAME	SOLUTIONS
YORK ID#	SOLUTIONS
PRISM LOGIN	SOLUTIONS

		Max points	Awarded points
(I)	1	10	
	2	10	
(II)		30	
(III)	1	10	
	2	10	
	3	10	
	4	10	
	5	10	
	6	10	
	7	10	
(IV)	1	10	
	2	10	
	3	10	
	4	10	
Total		160	

(I)

Assume that the following code fragments are executed. Display the values of the variables as requested below.

1. – **display the value of 'a' before the call and after the call of method AddNumber( ).**

```
void AddNumber ( ref int x)
{
     x += 33;
}
int a = 55;
AddNumber(ref a);
```

Answer: a = 55 (before call); a=88 (after call)

2. **display the value of 'a' after the call of method AddNumber( ).**

```
void OutNumber ( out int x )
{
     x = 55;
}
int a;
OutNumber ( a ) ;
```

Answer: a = 55 (after call)

(II) – write code

Assume we have a static method double `Sqrt(int x)` that returns the square root of a non-negative integer passed as parameter. Write a C# program that runs using a loop that asks the user to enter a number, and then calculates and prints the square root of that number. The program should stop when the user inputs 0 (zero). Also the program should not crash if the user enters a negative number. Use exceptions to achieve the latter.

Answer:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace Quiz01.II
{
    class Program
    {
        static void Main(string[] args)
        {
            while (true ) {

                Console.Write("enter number: ");
                string numberString = System.Console.ReadLine();
                int theNumber = Int32.Parse(numberString);

            if (theNumber != 0)
            {
                Console.WriteLine("you entered {0} ", theNumber);
                try
                {
                    if (theNumber < 0)
                    {
                        throw new ArithmeticException();
                    }
                    double result = Sqrt(theNumber);
                    Console.WriteLine("the square root is {0}",
result);
                }
                catch (ArithmeticException e)
                {
                    if (theNumber < 0)
                    {
                        Console.WriteLine("inside catch! you entered
{0}, a negative number", theNumber);
                    }
                    else
                    {
                        Console.WriteLine("exception caught: ",
e.ToString());
                    }
                }
            }
            else // entered zero ...
            {
                Console.WriteLine("you entered {0} .. program will
exit!", theNumber);
                break;
            }

        }
        }
        static double Sqrt(int x)
        {
            return Math.Sqrt(x);
        }

    }
}
```

(III) – circle one of ‘true’ or ‘false’

1. [true, [✓ false]] Properties must define get and set accessors.
2. [true, [✓ false]] Variables declared const may be initialized either in a declaration or in the class constructor.
3. [true, [✓ false]] Different namespaces cannot have classes/methods with the same names.
4. [[✓ true], false] Indexers can return any type in C#.
5. [[✓ true], false] Method ToString of class System.Object is declared as virtual.
6. [true, [✓ false]] A Car class has an “is a” relationship with its SteeringWheel and Brakes.
7. [true, [✓ false]] Exceptions can be thrown only by methods explicitly called in a try block.

(IV) – fill in the blank in each statement

1. Classes declared with keyword ____ sealed __________ cannot be inherited.
2. Class members can be overridden by another member M, if M uses the __ override _____ keyword.
3. To force an exception to occur when arithmetic overflow occurs in integer arithmetic, use operator __ checked ____.
4. Runtime exceptions derive from class ___ SystemException _____.