OOP 1 Exercises

1)

Install C# and make a “Hello World” console application.

2)

Write C# statements that accomplish the following:

Ask the user (Console.ReadLine) about her first- and lastname. Then greet her (Console.WriteLine) by her full name.

3)

Ask the user to enter a number. Use the appropriate method from the **System.Math** namespace to return the square root of the number.

4)

A) Write out the following via a loop (try both a for- and a while- loop):

*

**

***

****

*****

B) Write out the reverse, i.e., going from 5 down to 1 stars.

5)

1) Write out the number 300 in hexadecimal notation
2) Add the missing suffix to make the following assignments legal: `decimal d = 678.5;`
3) Add the missing integer data type to make the following assignment legal:
   d = 9990000000000000000;
4) Make a variable double d2 with the same value as d, but use exponential notation.

6)

Find fem fejl:

```
char a = "a";
bool b = 0;
int c = 8.0;
decimal d = 6.7;
string e = "Har du set "Holger"?";
```

7)

Create an enum type called PlayState with the following states: Play, Stop, Pause, Record. Start the numbering at 3.

8)

Create an array of lottery numbers. You can use

```
Random r = new Random();
r.Next(1, 43);
```

to generate random numbers.

9)

Make a list of the people in your group. Now, try adding and inserting family members, and try removing people again from the list.

10)

Ask the user to enter a number. Parse the number into an int. Try using both a Parse() version as well as a TryParse version. In the Parse() version, use appropriate exception handling.

11)

Experiment with the methods in the string class. For instance, given an input string, e.g., “the quick brown fox”, convert all white space to underscore ‘_’. Then try to split the string after each ‘_’ . [The Split() operation returns an array of strings which you can iterate over with **foreach** (string **in** string_array)].

13)

Create a library consisting of (at least) one method *IsSorted*. The purpose of *isSorted* is to determine

whether a collection (array or list) is sorted . That is, if you have just one element out of order, then the collection is, of course, not sorted. Make a version that checks collections of ints as well as a version that checks collections of string.

14)

A) Implement the following overloaded “Sum” methods. The methods should return the sum of the parameters.

static double Sum(double x1, double x2);

static double Sum(double x1, double x2, double x3)

static double Sum(params double[] numbers)

Try calling Sum(1, 2, 3) and Sum(1, 2, 3, 4). Which method gets called?

(You might want to try using the debugger, by placing a breakpoint at the call site, to see what is going on).

15)

- What get's printed in the Main() method below?
- One of the overloaded methods does not get called – call the missing M() method.
- Change the parameters of one of the M methods to make the x and y parameters optional.
- Add a method call in Main() where your method call makes use of named parameters.
- (this exercise is (pretty much) Example 49 in "C# Precisely" by Peter Sestoft and Henrik I. Hansen – a highly recommended reference book)

```
class Program
{
    static void Main(string[] args)
    {
        Console.WriteLine(M(false));
        Console.WriteLine(M(0));
        Console.WriteLine(M(3.0, 4));
        Console.WriteLine(M(3, 4));
        Console.WriteLine(M(3, 4.0));
        Console.ReadLine();
    }

    public static double M(int i) { return -i; }
    public static bool M(bool b) { return !b; }
    public static double M(byte x, byte y) { return x + y; }
    public static double M(int x, int y) { return 2 * (x + y); }
    public static double M(int x, double y) { return 3 * (x + y); }
    public static double M(double x, double y) { return 4 * (x + y); }
}
```

16)

- What gets printed in the Main() method below?
- The body of the last doStuff() method differs from the two others. Could we have written a *= 2 in the last method as well?

```
class Program
{
    static void Main(string[] args)
```

```
    {
        int b = 5;
        doStuff(b);
        Console.WriteLine(b);
        doStuff(ref b);
        Console.WriteLine(b);
        doStuff(out b);
        Console.WriteLine(b);
    }

    public static void doStuff(int a)
    {
        a *= 2;
    }

    public static void doStuff(ref int a)
    {
        a *= 2;
    }

    public static void doStuff(out int a)
    {
        a = 2;
    }
}
```