

1 C# Exercises

1.1 Get setup

- Download and install Mono.
- Write a Hello World program in C# (see below).
- Compile it with mcs, execute it with mono.

This enables you to play around and try the examples below.

1.2 HelloWorld.cs

```
using System;

class HelloWorld
{
    static void Main()
    {
        Console.WriteLine("Hello world!");
    }
}
```

Compile with mcs HelloWorld.cs

Run with mono HelloWorld.exe

1.3 Exercises – test yourself

Here are some questions so you can test your C# knowledge. These are exercises for yourself, you do not have to hand in your answers! Some of them are a little tricky, to help C++ programmers make the change! Some examples were taken from the book *Beginning Visual C# 2010* by Karli Watson.

1. (General) Do you know what .NET is, and what is Mono?

2. (Variables) Why do we have to define types of variables (in many programming languages)? Why is there more than one *numerical* type?

3. (Operators) Make sure you understand the C# operators. They are listed here: <http://msdn.microsoft.com/en-us/library/6a71f45d.aspx> – As a practise, what would the following code output?

```
int a = 5, b = 10, c = 5;

if ((a > 6 || b < 11) && c == 5)
    Console.WriteLine("test1");
else
    Console.WriteLine("test2");
```

4. (Namespaces) What is a namespace and what does the `using` directive do?

5a. (Branching) What's the result of this statement (ternary operator)?

```
int b = 5;
int a = (b < 10) ? 1 : 2;
```

5b. Make sure you understand the use of the `if` statement (see above example), and `switch`.

6a. (Looping) What does the following code produce?

```
for (i = 1; i <= 10; i++)
{
    if ((i % 2) == 0)
        continue;
    Console.WriteLine(i);
}
```

6b. The same process has been implemented using a `while` loop. But what is wrong with the code here?

```
int i = 1;
while (i <= 10)
{
    if ((i % 2) == 0)
        continue;
    Console.WriteLine("{0}", i++);
}
```

7. (Test) What does the following program do?

```
using System;

class Program
{
    static void Main(string[] args)
    {
        double balance, interestRate, targetBalance;

        Console.WriteLine("What is your current balance?");
        balance = Convert.ToDouble(Console.ReadLine());

        Console.WriteLine("What is your current annual interest rate (in %)?");
        interestRate = 1 + Convert.ToDouble(Console.ReadLine()) / 100.0;
```

```
Console.WriteLine("What balance would you like to have?");
targetBalance = Convert.ToDouble(Console.ReadLine());

int totalYears = 0;
while (balance < targetBalance)
{
    balance *= interestRate;
    ++totalYears;
}

Console.WriteLine("In {0} year{1} you'll have a balance of {2}.",
    totalYears, totalYears == 1 ? "" : "s", balance);

if (totalYears == 0)
    Console.WriteLine(
        "To be honest, you really didn't need to use this calculator.");

Console.ReadKey();
}
```

1.4 More advanced questions

Here are some more challenging questions, in case the above exercises were too easy.

8. (Variables) How can we convert from one variable type to another in C#?

9. (Value/reference) What's the difference between a value type and reference type in C#? What is boxing and unboxing?

CuuDuongThanCong.com

10. (Escape sequences) What will the following output in C#:

```
Console.WriteLine("Testing\noutput\tusing\0escape sequences");
```

11. (Operators)

Some operators are difficult to tell apart, e.g. & and &&

What is the difference between the & operator and && in the following if statements?



```
bool var1 = false, var2 = true;
```

```
if (var1 & var2)
    Console.WriteLine("AND1");
```

```
if (var1 && var2)
    Console.WriteLine("AND2");
```

What does the following code produce and why?

```
int a = 5, b = 6;
Console.WriteLine("a & b = {0}", a & b);
```

cuu duong than cong . com

cuu duong than cong . com