

Preparing for CM10192

Marina De Vos
mdv@cs.bath.ac.uk

1 Introduction

Welcome to the pre-course for the unit CM10192 or Programming. The aim of this introductory session is to give you an idea of what we will do and the tools we will use in this unit. Mainly we want to give you the opportunity to ease into the academic year and to obtain a head start.

The first year Programming unit is more than learning a programming language. We will look into problem solving techniques, provide you with the necessary concepts to teach yourself any programming language you want and to compare different ones to find the one most suitable for the problem you need to solve. We will also study basic algorithms to solve common problems, general design and implementation strategies. We will do this using two languages: Python and Java.

The pre-course will give you already a taster session. Hope you enjoy it. We have even thrown in a competition. If you have any questions, you can always reach me by email.

2 Unit Resources

For this unit we will use the following books:

- Python:
 - **How to Think Like a Computer Scientist: Learning with Python.** Allen B. Downey, Jeffrey Elkner and Chris Meyers. <http://www.ibiblio.org/obp/thinkCSPy/>
 - **Dive Into Python** Mark Pilgrim. <http://www.diveintopython.org/s>
- Java
 - **Thinking In Java.** Bruce Eckel. 3rd Edition. Prentice-Hall. 2002. ISBN: 031002872.
Free on-line copy at <http://www.mindview.net/Books/TIJ/>

You will need the following (free) software:

- Python 2.5 (<http://www.python.org>)

- Eclipse 3.2 (<http://www.eclipse.org>)
- Pydev plug-in (<http://www.python.org>)
- Java JDK 5 or higher (<http://java.sun.com>)
- BlueJ (<http://www.bluej.org>)

But most importantly, your time. Even with all the help in the world, you will not learn to program unless you do a fair amount of programming yourself. Or more to the point, you cannot pass this course by being passive. Compare it with a top football player or a dancer, they can never be good unless they exercise.

3 Finding Answers

The Internet has wealth of information that can assist you in finding solutions or provide easy answers to your questions. The tricky part is to find it, so good searching techniques are essential. To help you to get started we have added some exercises to find some tools and techniques that could prove to be useful for the unit.

- What is Emacs?
- Find installation instructions for Java JDK.
- Find an implementation of Quicksort in Java/Python.
- What are Makefiles? What is the Windows equivalent?
- Find a Java tutorial.
- What is an executable?
- What are binaries?
- What is Virtual Machine?
- What is Scratch?

4 Problem Solving

Problem solving is a crucial skill for computer scientist. Like any other skill, practice makes perfect. So the following exercises give you the opportunity.

For the following exercise, give a step-by-step (a bit like a cooking recipe) guide on how to solve the problem. Try to think about a solution yourself before you look on the Internet.

- Given a series of numbers in random order, you are asked to sort them in ascending order.

- Given a series of numbers in ascending order, you are asked to sort them in descending order.
- Given a series of numbers in random order, you are asked to find a specific number.
- Given a series of numbers in ascending order, you are asked to find a specific number
- Given two numbers X and Y, how would you print a tree with width X (9 in the example) and length X for the trunk (3 in the example)

```

      *
     ***
    *****
   *********
  ***********
   *****
    ***
     ***
      ***

```

- how would you solve the towers of Hanoi puzzle
- how would you solve the n-queens puzzle

5 Competition I: Scratch

Scratch (<http://scratch.mit.edu/>) is an easy programming environment developed by researcher of MIT. To let you experience that programming can indeed be fun, we decided to hold a little competition this year. Three boxes of Belgian chocolates can be won. They will be awarded for best concept, program difficulty and creativity. We have set up a gallery on scratch called "CM10192-Summer-Competition" so we can have a look at your achievements during the induction and give you an opportunity to see what others are doing. You can access it via <http://scratch.mit.edu/galleries/view/23958>. This gives you also an opportunity to get to know some of people you will share the course with. Good luck to all.

6 Python

6.1 Getting Started with Python

The following tasks have been designed to get you started with Python. Do not worry if not everything is clear to you. We will explain things in more detail during the lectures.

- Read the first chapter of "Dive into Python"
- Follow the instructions to install your favourite Python

- Run Python. To test type "print 1 + 1" and see what happens
- Read chapters 1-3 of the "How to think like a Computer Scientists" and section 2.1 of the "Dive into Python book".
- Try the exercises set in those chapters.

6.2 Some More Scripts

Time for some more serious stuff: functions.

- Download <http://www.cs.bath.ac.uk/~mdv/courses/CM10192/firstprogram.py>
- Have a look at the code. You should find 4 functions and one function call. Try to read to code.
- Run the code a few times. What happens if you enter positive numbers, negative numbers, characters, nothing?
- So what are all the functions doing?
- You were given a program without errors. This is perfect opportunity to see what happens if you make deliberate mistakes in the program. This will help you in the future when you make similar mistakes by accident.
 - remove spaces
 - remove :
 - change names of parameters
 - experiment

What does this tell you about the language?

- Now it is time to write some functions yourself.
 - Write a function that generates the third power a given number.
 - Write a function that takes three numbers as parameters and prints the smallest, biggest and average
 - Write multiplication of two numbers using only addition (e.g. $3 \times 3 = 3 + 3 + 3$)
 - Write functions *add_one* and *minus_one* that adds to or subtract one of its parameter
 - Implement a function (possibly with the assistance of other functions) that adds two numbers without the use of the + (except in *add_one*)
- The script you have just finished (or the original) can also be used as a module. This allows other scripts to use the functions defined in the module.

- This can be done with the import statement. Try this in your environment.
- To call the functions you will need to use "module.name.functionname" instead of functionname
- Try some function calls that way.

6.3 Eclipse

If you feel adventurous, download Eclipse and Pydev. Install eclipse and unzip/untar Pydev in the Eclipse directory. If you open eclipse (and taken the default option for the workspace) and go to Help followed by About, you get a window with a Python symbol on the left. This means you have successfully installed Pydev. Now go to Window followed by Preferences. Select Pydev on the left and then Interpreter-Python. On the right press the top most "new" button. When requested enter the path to your python interpreter and follow the instructions (defaults should be fine). Now you are set to use the Eclipse IDE for developing your python programs. Via File and New create a Python project. This will store all scripts and modules. Next, go to Window and Open Perspective to select Pydev. Now you can just open a new file, give it a name with extension .py and you start to program. When you are ready to run, select the file on the left, right click and select Run As and choose Python run. You will see the result appear in your console view.

7 Java

7.1 Getting Started with BlueJ

The following tasks have been designed to get you accustomed to BlueJ and some terminology of Java. Do not worry if not everything is clear to you. We will explain things in more detail during the lectures.

- Read the first chapter of Thinking in Java
- Download the BlueJ Tutorial (<http://www.bluej.org>).
- Read Sections 1 and 2.
- Install BlueJ and if necessary Java SDK (<http://java.sun.com>).
- Read sections 3,5,9.1 and 9.2. While doing so, practise their suggestions in BlueJ.

7.2 Your First Programming Steps

Now that you are used to BlueJ, it is time to do something programming.

- Download FirstClass.java <http://www.cs.bath.ac.uk/~mdv/courses/CM10192/FirstClass.java>

- Import the code into BlueJ
- Have a close look at the code: read the comments (the English text with stars and slashes)
- Create a FirstClass object as seen before.
- Test all three the methods
- Did you notice the difference between a return value and no return value?
- The code you received did not contain any error. However, in your programming it is most unlikely that you will never make any mistakes. To prepare yourself for it and to see how the java compiler reacts to it, it is important that you make mistakes in purpose. Try the following things one at a time:
 - delete a semicolon somewhere
 - Change the name of the class
 - Change the name of a constructor: For example change to lower case for Class
 - In public int firstMethod, change M to m
 - In public String secondMethod, replace String with int
 - In public String secondMethod, delete String
 - Delete return result in one of the methods
 - In thirdMethod, replace firstMethod(three,four) with firstMethod(three)
 - In thirdMethod, replace secondMethod(one,two) with secondMethod(one,three)
 - delete {
- Now it is time that you write something yourself. Write a method in FirstClass that is capable of computing the average of four numbers. Use double instead of int to allow decimals. Inform both the user and the program of your result.
- Documenting your code is essential for programmers. You want other to be able to understand what you are doing. Therefor, document your method in a similar style as the other methods.
- Find out what would happen if you would change the doubles you used by int.

7.3 Code Prediction

Being a programming implies that you do not only have to write your own code, but that you must be able to understand code written by other.

- Download the code of SecondClass <http://www.cs.bath.ac.uk/~mdv/courses/CM10192/SecondClass.java>.
- Import it into BlueJ.

- Unfortunately you just received badly documented code. Nevertheless you have to find out what the code does. In this case, the only option is to use trial and error. Run the code a couple of time to find out what it is doing.
- Document the code so that everybody is able to understand the code.
- Adapt the code of `secondMethod` in such a way that the sequence is printed out in reverse order.
- Write a method that generates the third power a given number.
- Write a method that generates a sequence of all third powers of odd numbers.

7.4 Theory

You have read the first chapters of the accompanying books and you have been playing with classes and objects in BlueJ, so by now you should have some vague idea what the difference is between the two.

- Identify the classes and the associated objects in the following collection: “Harry Potter and the Philosopher’s Stone”, City, `theFraction3/4`, Bath Student, Book, CM10192, BankAccount, you, Fraction, CM50176.
- Describe the features of each class and identify the actual valued for the corresponding objects. For example the class Cat has a feature name. The object `myCat` has a value “Vampire” for name.
- What should you be able to do with each class/object. For example, ask a Cat’s name.
- Find some other example of classes and objects.

7.5 Small Challenges

For those of you who already want to do a bit more programming.

- Write a piece of code, well-documented, in BlueJ to compute the surface of a circle. Hint: Have look in the Java API for the Math class
- Write a piece of code, well-documented, in BlueJ that checks if one String is a substring of an other.
- Write a piece of code, well-documented, in BlueJ that picks all the words out of a sentence. Hint: browse the Java API.
- Write a piece of code, well-documented, in BlueJ that reverses a String.

7.6 Python vs. Java

Now that had some programming experience in both languages, give some similarities and difference between the two languages. Which one did find easier to use?