Principles of Programming SS, 2019

Course Information

COMPSCI 101 – PRINCIPLES OF PROGRAMMING

This course is an introduction to programming computers. It is the main introductory course in the Computer Science department and is taken by students from a variety of disciplines wishing to have an understanding of computer programming as well as students wanting to continue on to further studies in Computer Science.

We teach programming using the cross-platform language Python. The main focus is on learning to understand the detailed requirements of a programming task, and writing programs that are well structured, correct, easy to read, and to maintain. In order to do this students need to develop an understanding of how to represent information both as data and algorithms. Students also need to develop the skills of incrementally developing and testing programs.

The course covers simple variables, expressions, input and output, control structures, functions, using standard data structures such as lists and dictionaries, and using standard Python modules.

By the end of the course students who succeed should be able to design and implement a medium-size computer program as well as have some idea of the process of program execution.

LEARNING OUTCOMES

A student who successfully completes this course will be able to: understand code, implement algorithms, test code, document code, design solutions using functional decomposition and implement those solutions. More specifically, a student who successfully completes this course will be able to:

- determine the state of the program both during and after execution, given a code listing that may include functions and parameters, loops, conditionals and sequences
- implement a given algorithm using Python
- show that a program meets given specifications by writing appropriate tests
- provide a useful level of documentation, in the form of program comments, for all programs developed
- decompose a simple problem into several smaller tasks, given a brief textual description of the problem
- compose functions that perform a specified task into a program that solves a given problem
Programming is traditionally viewed as a subject in which concepts build progressively, so each new concept requires a detailed understanding of the concepts learned previously. Although a number of programming concepts are interrelated, it is possible to separate many of the ideas and to introduce new concepts in a way that relies on little previous knowledge. Additionally, many texts use an approach in which each topic is explored deeply before moving onto the next topic.

This course attempts to present material using a layered approach, in which a broader range of topics is covered initially in a shallow way, then revisited in more depth at later stages. This approach provides a more gradual exposure to the core ideas and allows learning to continue when a particular topic is not well understood.

The programming concepts are introduced primarily through the use of examples. Students are strongly encouraged to practise these examples by typing them into a computer and modifying them to get some hands-on experience.

---

**Teaching Staff**

Below is the contact information of the three people involved with the teaching of CompSci 101:

**Ann Cameron (Lab Supervisor)**
- **Room:** 413, Maths Building (Building 303)
- **Phone:** (09) 9234947
- **Email:** ann@cs.auckland.ac.nz

**Adriana Ferraro (Course Coordinator)**
- **Room:** 415, Maths Building (Building 303)
- **Phone:** (09) 9237113
- **Email:** adriana@cs.auckland.ac.nz

**Joerg Wicker (Lecturer)**
- **Room:** 526, Maths Building (Building 303)
- **Phone:** (09) 9232184
- **Email:** j.wicker@auckland.ac.nz
## CompSci 101, Summer School, 2019

<table>
<thead>
<tr>
<th>Day</th>
<th>Lecture Schedule</th>
<th>Labs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Monday</td>
<td>7-Jan  1 Introduction</td>
<td></td>
</tr>
<tr>
<td>Tuesday</td>
<td>8-Jan  2 Variables, program execution, doing calculations, print()</td>
<td>Where is the lab?</td>
</tr>
<tr>
<td>Wednesday</td>
<td>9-Jan  3 Expressions, mathematical operators, the math module</td>
<td></td>
</tr>
<tr>
<td>Thursday</td>
<td>10-Jan 4 The type() function, strings, objects, the len() function, string slices</td>
<td></td>
</tr>
<tr>
<td>Friday</td>
<td>11-Jan 5 Manipulating strings, string methods, dot notation</td>
<td>LAB 1</td>
</tr>
<tr>
<td>Monday</td>
<td>14-Jan 7 Defining functions 1</td>
<td>LAB 2</td>
</tr>
<tr>
<td>Tuesday</td>
<td>15-Jan 8 Defining functions 2, functions can call other functions, scope of variables</td>
<td></td>
</tr>
<tr>
<td>Wednesday</td>
<td>16-Jan 9 Code tracing, divide a program into separate tasks</td>
<td></td>
</tr>
<tr>
<td>Thursday</td>
<td>17-Jan 10 Boolean expressions, if statements, equality and floats</td>
<td></td>
</tr>
<tr>
<td>Friday</td>
<td>18-Jan 11 if ... else, if ... elif statements, nested ifs</td>
<td>LAB 3</td>
</tr>
<tr>
<td>Monday</td>
<td>21-Jan 13 The Python range() function, for ... in loops</td>
<td>LAB 4</td>
</tr>
<tr>
<td>Tuesday</td>
<td>22-Jan 14 Lists, for in loops to iterate through the elements of a list</td>
<td></td>
</tr>
<tr>
<td>Wednesday</td>
<td>23-Jan 15 The split() method, lists are mutable objects</td>
<td></td>
</tr>
<tr>
<td>Thursday</td>
<td>24-Jan 16 Slicing lists, list methods</td>
<td>LAB 5</td>
</tr>
<tr>
<td>Friday</td>
<td>25-Jan 18 Revision (TEST IS ON ALL THE MATERIAL UP TO THE END OF lecture 15)</td>
<td>A2 (Thursday)</td>
</tr>
<tr>
<td>Monday</td>
<td>28-Jan Auckland Anniversary Day</td>
<td>No Lecture</td>
</tr>
<tr>
<td>Tuesday</td>
<td>29-Jan No Lecture</td>
<td>No Lecture</td>
</tr>
<tr>
<td>Wednesday</td>
<td>30-Jan TEST</td>
<td></td>
</tr>
<tr>
<td>Thursday</td>
<td>31-Jan 19 Open files, read from files, write to files, close files</td>
<td>LAB 6</td>
</tr>
<tr>
<td>Friday</td>
<td>1-Feb 20 Maintaining a text file of information</td>
<td></td>
</tr>
<tr>
<td>Monday</td>
<td>4-Feb 21 Dictionaries</td>
<td>LAB 7</td>
</tr>
<tr>
<td>Tuesday</td>
<td>5-Feb 22 More on dictionaries</td>
<td>A3 (Tuesday)</td>
</tr>
<tr>
<td>Wednesday</td>
<td>6-Feb 23 tkinter 1 - GUI's, using the Canvas widget</td>
<td></td>
</tr>
<tr>
<td>Thursday</td>
<td>7-Feb 24 tkinter 2 - Drawing shapes using nested loops</td>
<td>LAB 8</td>
</tr>
<tr>
<td>Friday</td>
<td>8-Feb 25 More on nested loops, passing mutable objects as parameters</td>
<td></td>
</tr>
<tr>
<td>Monday</td>
<td>11-Feb 26 Using the Python interpreter, Python sequences</td>
<td>LAB 9</td>
</tr>
<tr>
<td>Tuesday</td>
<td>12-Feb 27 Testing, docstrings for functions, doctests</td>
<td>A4 (Monday)</td>
</tr>
<tr>
<td>Wednesday</td>
<td>13-Feb Revision for the CompSci 101 Exam</td>
<td></td>
</tr>
<tr>
<td>Thursday</td>
<td>14-Feb No Lecture</td>
<td></td>
</tr>
<tr>
<td>Friday</td>
<td>15-Feb No Lecture</td>
<td></td>
</tr>
<tr>
<td>Saturday</td>
<td>No Lecture</td>
<td></td>
</tr>
<tr>
<td>Monday</td>
<td>18-Feb Exams (Feb 18 - Feb 20)</td>
<td>A5 (Sunday)</td>
</tr>
</tbody>
</table>
Your final grade will consist of 30% practical components and 70% theory components. The theory component will consist of a test worth 10% and a final exam worth 60%. The practical component will consist of 9 labs worth 13.5% and 5 assignments worth 16.5%.

This course is designated as being of a practical nature. This means that you must pass both the practical component (labs and assignments) and the theoretical component (test and exam) separately.

**Laboratories**

The laboratories are worth 13.5% of your final mark. The labs start on Thursday January 10th/Friday January 11th. Attendance at the labs is compulsory. Every laboratory session contributes towards your final grade. Please keep your signed lab sheet as proof of lab attendance.

The labs are designed to give you practical experience with the basic concepts which you have learnt in lectures. They will include activities to be completed during the laboratory session.

Each week you should make sure you have read through the lecture slides and have done any other preparation required before you attend the laboratory.

**Assignments**

Assignments are designed to give you practical experience with the concepts which you have learnt in lectures. Assignments are worth a total of 16.5% of your final mark. There are five assignments and each is due at 4:30pm on their due date. The first assignment is due on Friday, January 18.

For three of these five assignments (a total of 10.5%) you are required to write one or more programs. These three assignments are to be submitted to the assignment dropbox:

https://adb.auckland.ac.nz/Home/

The other two assignments (a total of 6%) use the CodeRunner2 tool. This tool is designed to help you practise by presenting you with a set of short online exercises. Submissions are graded by running a series of test cases of the code in a sandbox and comparing the output with the expected output. You will be able to access your exercises by logging into

https://www.coderunner2.auckland.ac.nz/moodle/

**Test**

The test is worth 10% of your final mark, and will be held during the lecture time (9:05am – 10:25am) on Wednesday January 30. The location of the test will be announced during lectures closer to the time. If you are enrolled in another course that has a test scheduled for the same time (or if you are unable to attend this test time for another valid reason), then please contact the course coordinator, Adriana Ferraro, as soon as possible.
Exam

The final exam is worth 60% of your final mark. Please check Student Services Online for the exam time and date. The exam is closed book, and calculators are not permitted. Provisional exam results can be obtained from Student Services Online.

OTHER INFORMATION

How to seek assistance

In the labs, there are always tutors and demonstrators available to help you. If you have an administrative problem (e.g. you have been ill, you have a timetable clash with your lab or test, your marks have been incorrectly recorded, etc.), or any other sort of problem that you need help with, please see the course coordinator. Students are asked to discuss privately any impairment related requirements face-to-face and/or in written form with the course coordinator or lab supervisor. If you need extra help with understanding the course material, or preparing for the test or exam, you are very welcome to visit any of the teaching staff either during their office hours or at some other time when they are available.

There are many other resources available within the University, e.g. the Student Learning Centre, the library, DELNA (to identify where you may need help with your academic English) and ELE (English Language Enrichment - a set of resources to help you improve your English).

Missed exam

If you miss the exam for any valid reason, or you sit the exam but believe that your performance was impaired for some reason, then you may be able to apply for an aegrotat, compassionate or special pass consideration. For more detailed information, refer to the University of Auckland’s Calendar.

Webmail

All students have a university email account. Your university email address is: NetID@aucklanduni.ac.nz, e.g., abcd001@aucklanduni.ac.nz. You can access your email from anywhere you have Internet access, by logging into

http://webmail.ec.auckland.ac.nz

You must read email sent to your university email address regularly, as staff members often send important messages to students via their university email address. When emailing staff members, please use your university email address.
Course Information

The CompSci 101 Website

On the COMPSCI 101 website you will find course information, assignments and lecture notes:

http://www.cs.auckland.ac.nz/courses/compsci101ssc/

Lecture Recordings

Recorded lectures can be found on Canvas.

Checking your marks on Canvas

You can check your marks by logging onto Canvas. If there are any problems with your lab marks or test marks, please contact Ann Cameron. If there are any problems with your assignment marks, please contact Adriana Ferraro.

Policy on Cheating and Plagiarism

Cheating is viewed as a serious offence by the University of Auckland. Penalties are administered by the Discipline Committee of the Senate, and may include suspension or expulsion from the university. Do not copy anyone else’s work, or allow anyone else to copy from you.

For more information on the University’s policy on cheating, please refer to the web page:

http://www.auckland.ac.nz/uoa/home/about/teaching-learning/honesty

Do not ever copy anyone else’s work, or allow anyone else to copy from you.

Piazza

The Piazza discussion forum within Canvas is regularly monitored by teaching staff. Please make use of the forum to ask any questions that you think might be of interest to other students. If your question is of a personal nature, or relates to a unique situation that will be of little interest to others, then please contact the teaching staff directly.
Print Quota

You can add credit to your print quota at the library or the IC Helpdesk on Level 2 of the Kate Edger Information Commons, 11 Symonds St.

Your first lab

Lab sessions start on Thursday January 10th/Friday January 11th. When you arrive at the lab, you should sign next to your name on the lab noticeboard. There will be tutors and lab demonstrators available throughout all the labs to help you. In order to use any of the computers you will need to log into the system. This will be the username and password which you use to log into Student Services Online.

Please bring your Student ID card and a USB Flash drive to every lab.

Label all your flash drives on the outside. It is also sensible to have a file called MY CONTACT DETAILS.txt with your contact details stored on your flash drive as several flash drives are left in the lab each week.

Enjoy the course!