Computing Science 120 • Study Guide

Introduction to Computing Science and Programming I

Fall 2010 Edition

by

Greg Baker

Faculty of Applied Sciences
Simon Fraser University

Contents

List of Figures

Course Introduction

[Most of the material in this introduction will apply to all offerings of CMPT 120, but some details will vary depending on the semester and instructor. Please check your course web site for details.]

Welcome to CMPT 120, "Introduction to Computing Science and Programming I". This course is an introduction to the core ideas of computing science and the basics of programming. This course is intended for students who do not have programming experience. If you have done a significant amount of programming, you should take CMPT 126 instead.

This course isn't intended to teach you how to use your computer. You are expected to know how to use your computer: you should be comfortable using it for simple tasks such as running programs, finding and opening files, and so forth. You should also be able to use the Internet.

Here are some of the goals set out for students in this course. By the end of the course, you should be able to:

- Explain some of the basic ideas of computing science.
- Explain what computer programming is.
- Create algorithms to solve simple problems.
- Implement computer programs in the Python programming language.
- Apply the core features of a programming language to solve problems.
- Design programs that are easy to understand and maintain.

Keep these goals in mind as you progress through the course.

Learning Resources

Study Guide

The *Study Guide* is intended to guide you through this course. It will help you learn the content and determine what information to focus on in the texts.

Some suggested readings for each section are listed at the beginning of the units. The instructor for your section of the course may modify these readings. You should also look at the key terms listed at the end of each unit.

In some places, there are references to other sections. A reference to "Topic 3.4," for example, means Topic 4 in Unit 3.

◈ In the *Study Guide*, side-notes are indicated with this symbol. They are meant to replace the asides that usually happen in lectures when students ask questions. They aren't strictly part of the "course."

Required Texts

The only required text for this course (besides this *Study Guide*) is *How to Think Like a Computer Scientist: Learning With Python.* It is available from the SFU bookstore, or you can download it from the course web site. This book is a nice introduction to Python programming, which we will use in the second half of the course.

The title of the book is a little misleading. The book does *not* discuss computer science; it only covers computer programming. In this Guide, we will try to cover more actual "computing science."

Recommended Texts

There are currently no recommended texts for this course. If additional resources are required during the semester, the instructor may recommend other books and place them on reserve at the library.

Web Materials

The web materials for this course can be found in Computing Science's "Course Central," http://www.cs.sfu.ca/CC/ .

Online References

There are several online references that are as important as the texts. You can find links to them on the course web site.

These resources are very important to your success in this course. They aren't meant to be read from beginning to end like the readings in the textbook. You should use them to get an overall picture of the topic and as references as you do the assignments.

Email Communications

The TAs and course instructor will use email to send announcements and tips during the semester. You should read your SFU email account regularly, at least a few times each week.

You can also contact the TAs and course instructor by email if you need help. See the "Getting Help" section at the end of the Introduction for details.

Requirements

Computer Requirements

You need to have access to a computer for this course. The labs on campus or your own computer can be used. Instructions on using the labs can be found on the course web site.

You will also need a connection to the Internet through a dial-up connection (one is provided with your SFU registration) or through another type of connection like a cable modem or a DSL line. A high-speed connection will make your life easier, but it isn't required.

You should activate your SFU computing account as soon as possible if you haven't done so already. Instructions can be found in your registration material.

Software Requirements

All of the software that you need for this course can be downloaded for free. Links to obtain the course software are on the course web site. Appendix A contains instructions for installing and working with the Python software.

There may be some other pieces of software that you'll need to use when submitting your work. See the course web site for instructions on installing and using these.

Evaluation

Marking

The marking scheme for the course will be announced in class or by email in the first week.

Labs and Assignments

You will have weekly labs in this course. Details will be announced in class. The labs will consist of relatively short exercises that you should be able to complete quickly. They are intended to make sure you stay up to date with the material in the course as it progresses.

The assignments are more involved than the labs. You will have to figure out more on your own and explore the concepts on the course.

Exams

There will be two examinations in this course. They are closed-book, i.e. you aren't allowed any notes, calculators, or other aids. The exams have a mixture of short and longer answer questions.

The 50 minute midterm exam will be held in week 7 or 8. The instructor will announce what material will be covered in the midterm a few weeks before. The final exam will be three hours long and will cover all of the course material.

Academic Dishonesty

We take academic dishonesty very seriously in the School of Computing Science. Academic dishonesty includes (but is not limited to) the following: copying another student's assignment; allowing others to complete work for you; allowing others to use your work; copying part of an assignment from an outside source; cheating in any way on a test or examination.

If you are unclear on what academic dishonesty is, please read Policy 10.02. It can be found in the "Policies & Procedures" section in the SFU web site.

Cheating on a lab or assignment will result in a mark of 0 on the piece of work. At the instructor's option, further penalties may be requested. Any academic dishonesty will also be recorded in your file, as is required by University policy.

Any academic dishonesty on the midterm or final will result in a recommendation that an F be given for the course.

About the Author

The author hates writing about himself, so he isn't going to. So there.

I'd like to thank Arsalan Butt, Yaroslav Litus and Mashaal Mamemon who worked on the creation of this course with me in the summer of 2004. Thanks also to Toby Donaldson, Anne Lavergne, David Mitchell, Brad Bart, John Edgar, and all of the other faculty members around the School that were willing to listen to an idea and, more often than not, reply with a better one.

If you have any comments on this guide, please feel free to contact me at ggbaker@cs.sfu.ca.