1 Introduction

Welcome to 1001ICT Introduction To Programming (ITP) and 7001ICT Programming Principles 1 (PP1).

These lecture notes are divided into sections. This one describes the course’s:

• content;

• roles and responsibilities;

• assessment;

• resources; and

• some things you need to do now.
This course is being offered in parallel at the:

- Logan campus; and
- Nathan campus.

It is the same course on both campuses, just different lecturers.
1.1 7001ICT Programming Principles 1

For this semester, the class also includes Masters students enrolled in 7001ICT PP1.

The course materials and assessment items will be the same as for the undergraduate course, however the grading basis will be different.

Please note that this is an introduction to programming.

If your undergraduate program included programming, please see the course convenor or your program convenor for advice as to whether this is the right course for you.
1.2 What to do in lectures

1.2.1 Why have lectures?

• They are an efficient way to convey the content of a course.

• The meaning of the content is conveyed better verbally than just in a written form as:
  
  – It’s less boring than just reading.
  
  – Some things are hard to read by yourself at first (programs, maths, diagrams).
  
  – It’s easier to tell what’s important from the emphasis given.
  
  – You can ask questions.
  
  – It’s more social.
• It’s regular time set aside for thinking about the course.

• It’s the best place for important announcements.

• It’s an opportunity for feedback about how the course is going and how the class is performing.

It’s harder to keep up in a course if you are missing lectures.
1.2.2 Take notes?

These notes are posted on the web.
So you don’t have to copy them all down.
But we say more than goes in the slides.
You should be prepared to take notes that capture the extra information.
Lecture notes are available on the web before the lecture, so bring a printout,
so you can add extra highlighting or comments.

1.2.3 Take notes today!

Write down the things in colour.
1.3 Course Convenors & Lecturers

Logan lecturer

Ali Zia
ali.zia@griffithuni.edu.au

Nathan lecturer & course convenor

Andrew Rock
a.rock@griffith.edu.au
Technology Building (N44) 1.37
1.4 Course web site

These notes will be posted on the web at:
http://www.ict.griffith.edu.au/arock/itp/

You will find the laboratory notes and anything else you need there too.

Write down the uniform resource locator (URL) now!

You can also access these resources via Learning@Griffith, though direct access via the above URL will be quicker.

Nathan lectures are captured and available, after editing, on Learning@Griffith’s Echocentre.
1.5 What the course is about

This is an introduction to:

- programming tools;
- programming concepts;
- problem solving; and
- programming languages.

A lot of the programs we will write will be for robots.
Others will be for personal computers, with text or graphical output.
We will be learning and programming in a subset of the Java programming language.
1.6 Why learn programming

The applications we use, the internet, games, entire businesses are made up of programs, and someone has written them.

The ability to create programs is important if you want to participate in building the future.

The understanding of what can and can’t be achieved with a program is important, even if you don’t write programs yourself. People are needed to design, document and specify programs, as well as code them.

Multimedia students should particularly consider programming experience as their edge over competitors doing ordinary design courses.
Many courses will tell you that they will help you develop problem solving skills. (Employers like it. “Don’t come to me with problems. Come to me with solutions!”)

But a program is usually not the specific solution to one instance of a problem, it is the solution to a whole class of problems, written out clearly. That is a higher level of problem solving.

Programming is one of those things that is satisfying, enjoyable and rewarding “... not because they are easy, but because they are hard...” (JFK).
1.7 The choice of programming languages

In your working life you will write lots of programs, in lots of programming languages.

Any time you create a script, automation, spreadsheet, or animation, it’s programming.

It may be that you never program in Java after completion of your programming classes.

However, Java is a programming language that has all of the attributes found in all of the common application and scripting languages. Its notation is similar to many, including C, C++, C#, JavaScript, and ActionScript.
1.8 Can you learn programming?

Yes!

*Easy*: Convincing us you are capable of programming if you try. We already believe it.

*Hard*: Convincing yourself.

Come to classes. Pay attention. Don’t panic. Persevere. Ask questions. You will pass this course.
1.9 Classes and assessment

1.9.1 Lectures

In a 2-hour lecture every week, we will learn the concepts and strategies used in programming.

During 5 of the lectures, we will test your learning with quizzes.

\[ 3 + 4 + 5 + 6 + 7 = 25 \text{ marks} \]

1.9.2 Workshops

In a 1-hour workshop every week we demonstrate more examples and provide and discuss extra non-assessed problems.
1.9.3 Labs

In 10 2-hour assessed lab classes you put the theory into practice.

\[ 2 + 2 + 3 + 3 + 4 + 4 + 5 + 5 + 6 + 6 = 40 \text{ marks} \]

1.9.4 Common times

A common time (we call them a “drop-in”) is a class where there are no set activities, but you can get individual help if you need it or we can challenge you with extra, harder problems if you want them.

1.9.5 Final project

The course ends with a larger programming project (35 marks).
1.9.6 Attendance summary

Students are expected to attend all lectures and workshops.

You are enrolled in one 2 hour labs class. Attend that every week from week 2.

Attend any common time as needed, from week 2.

“Eighty percent of success is showing up.” – Woody Allen.

1.9.7 Assessment summary

<table>
<thead>
<tr>
<th></th>
<th>3 + 4 + 5 + 6 + 7 =</th>
<th>2 + 2 + 3 + 3 + 4 + 4 + 5 + 5 + 6 + 6 =</th>
<th>35</th>
</tr>
</thead>
<tbody>
<tr>
<td>quizzes</td>
<td></td>
<td></td>
<td>25</td>
</tr>
<tr>
<td>labs</td>
<td></td>
<td></td>
<td>40</td>
</tr>
<tr>
<td>project</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>total</td>
<td></td>
<td></td>
<td>100</td>
</tr>
</tbody>
</table>
1.9.8 Minimum requirements for labs and quizzes

The labs are quizzes are treated as one assessment item each.

If you do not attend the majority of each, we don’t think you are really attempting them.

If you don’t attend at least 6 labs, you don’t get any lab marks at all.

If you don’t attend at least 3 quizzes, you don’t get any quiz marks at all.

Either of these outcomes will make it almost impossible to pass the course.

You may even be awarded a grade of Fail Not Sat, which would be a better indication of your performance than a plain Fail.
1.9.9 Supplementary exam

Supplementary assessment, if passed, can raise your grade from a 3 to the lowest passing grade (a 4).

To be eligible, you must have been awarded a 3, have submitted/sat the majority of the assessed labs, the majority of the quizzes, and at least part of the final project.

For this course, the supplementary assessment takes the form of a two-hour, closed-book, written examination.
1.10 Lecture notes

The lecture notes are available on the course web site in various formats. The most useful format is the A5 version, which:

- has a table of contents;
- has a glossary and an index;
- has clickable links;
- includes appendices that summarise the programming language and environments, plus other useful reference information;
- is ideally formatted for a tablet or laptop screen; and
- prints double-sided as a small book.
1.11 Griffith Sciences Laboratory Induction

You will have been briefed in orientation about workplace health and safety, and the on-line course and test on Learning@Griffith in the organisation Griffith Sciences Laboratory Induction that you must complete.

In this course we will be checking that you have passed that test.

You must print the certificate that shows that you passed that test, *within the last year*, and bring it to your first programming lab class (in week 2).
1.12 Peer Assisted Study Sessions

This course has available a student learning support service, Peer Assisted Study Sessions (PASS).

PASS offers free extra weekly study sessions, led by more experienced fellow students (PASS leaders).

The details about PASS will be presented by the PASS leaders in the week 1 lecture.

PASS is optional (and limited access), but students who use it have been shown to get better results than students who don’t.
1.13 Laboratory 1

The first lab will be held in week 2.

The exercise notes are on the course web site.

We will find and preview them now.

We will also find the Griffith Sciences Laboratory Induction on Learning@Griffith.
1.14 Section summary

This section covered:

• what happens in lectures;

• why this course on programming is an important part of your program of studies;

• how this course is organised, taught and assessed;

• where the resources for this course can be found on the web;

• who the teachers are and how to contact them; and

• what you need to do to get started in this course.
1.15   End of section feedback questions

Send us your answers to these questions any time you like by clicking on them.

• What was the most useful topic in this section?
• What was the least useful topic in this section?
• What was the least clear topic in this section?
• What topic in this section would you like to know more about?
• Did you find an error in this section?
1.16 Things to do

What you need to be doing as soon as possible to get started:

- Get a diary or organiser and plan how you can spend your week profitably, including the following items.

- Make sure you are properly enrolled in this course and in a laboratory.

- Read the course outline and the course web site.

- Prepare for laboratory 1, in week 2. Find the notes on the web site. There are some questions you need to answer before your lab class.

- Complete the Griffith Sciences Laboratory Induction on Learning@Griffith, and bring the printed certificate to labs at least once.
• Lecture notes appendices A through G contain useful information that you should read now.

• Prepare for the first workshop (in week 2) by reading the exercises on the web site when they are released.