SCIE1000/SCIE1100 Tutor Guide, 2017

1. Introduction

Welcome to your role as a tutor for *SCIE1000, Theory and Practice in Science*. SCIE1000 covers a range of content from mathematics, science, programming and philosophy of science. Unlike many other courses, the primary goal of SCIE1000 is to get students to think in different ways, rather than to cover deep discipline-based content, and we see a range of evidence that we are fairly successful at this. The course scores highly on student feedback, and is regularly identified by high achieving students as having a significant impact on how they are thinking. Of course, some students do not like or value the course, feeling that it may be too easy or too hard for them, or preferring to encounter discipline-based content rather than interdisciplinary thinking. One of the challenges for you and for us is to try to minimise the number of students who respond to the course and its content in that way, by maximizing student engagement and learning.

We are proud of the rationale for SCIE1000, along with its content, the level of support provided to enrolled students, and the teaching team. This team comprises lecturing staff from a number of faculties and all of the tutors. All lecturers and tutors play important roles in the success of SCIE1000 and the enrolled students, and we are very pleased that you are able to participate this year. We hope that you will find your involvement with the course as fulfilling as we do!

(Your tutorial classes will comprise students enrolled in SCIE1000, and may also include some students enrolled in *SCIE1100, Advanced Theory and Practice in Science*. Most of the time, it does not matter in which course a student is enrolled. SCIE1100 students attend all SCIE1000 lectures and tutorials, along with an additional, specialized class each week. Except when otherwise noted, all mention of SCIE1000 in this document should be taken to include SCIE1100 as well.)

2. Key changes for 2017 that will impact on tutors

If you have tutored SCIE1000 before or previously completed the course, please pay particular attention to the following changes from previous years. **In particular, we have made the largest changes to Python and programming since the course was first offered ten years ago.**

a) We have written this guide… Make sure you read it and pay attention!

b) A number of science computer labs are no longer Apple Macs running in dual boot mode, but instead are PCs running windows. This should simplify things for students and tutors. Before your first class, make sure you can login and that you understand how things work.

c) We have changed the version of Python we use, to a more recent version, from a different developer. We now use *Anaconda 3.5.x*. The screen looks completely different, and is more user friendly than in the past. You **must** go to a computer laboratory prior to your first class and be sure you know how to run the correct version of Python. The link is called *Jupyter Notebook* and is accessible from the start button. Be clear on how to start and stop the program, where new programs are written, how to run a program, where the output will appear, where students should save their files, and so on. *Jupyter Notebook* also includes a "smart" editor, which aids in formatting the program commands.

d) As part of the change to using *Jupyter Notebook*, all tutorial sheets are now presented as iPython Notebooks. This means that the Python, mathematics and science questions are all integrated into single workbooks, with links to internet resources, and programs run 'in-situ', and output embedded within questions. **It's very important that you familiarize yourself with this new approach to tutorial sheets.**

e) As part of the change in Python version, there are a few differences to programming commands:

   i. The line "from __future__ import division" is no longer used. (This version of Python automatically does integer division in the standard, mathematical manner.)

   ii. The "print" command now uses parentheses around the things to be printed, as with any function call (the "print" command is actually a Python function). An example is: *print("Hello ", 4*5, "sailors")*.

   iii. The "input" command now needs to also use the "eval" command. An example is: *mass = eval(input("What is your mass?"))*.

   iv. Graph plotting is slightly different, with graphs appearing even without the "show" command, and multiple graphs able to be displayed on the screen at the same time.

f) We intend to include a midsemester exam with a higher percentage value than in the past.

g) It is likely that SCIE1000 will become compulsory in the BSc from 2018, so we are extra keen for the 2017 offering to work well, and be useful to students. Please offer any good advice you might have, to us and to fellow tutors.

3. Our expectations of you

When you undertake a paid role within the University, you have changed from being a student to being a staff member. The University has a range of professional expectations of all of its staff members, and these are outlined in UQ policies. You are bound by these codes of behavior. You also take on a number of responsibilities, such as Occupational Health and Safety, being a member of a diverse community, and showing respect for colleagues and students. If you wish to read more, you could consider visiting www.uq.edu.au/current-staff, and exploring.

We have a range of expectations specifically relating to how you approach your role as a tutor in SCIE1000. For example, we expect you to read this tutor guide, be familiar with its contents, ask questions if anything is unclear or you have any concerns, to and make suggestions as to how we can improve any aspects of the course.

As a tutor, you must:

- Follow any reasonable instructions we give you, or requests we may make of you.
- Ensure that you are academically prepared for each of your classes.
- Take your teaching role seriously, whilst being engaging to your students.
- Inform us of any problems that may arise, or concerns you might have.
- Think about different ways to explain difficult concepts.
- Show respect for all of the course teaching team.
- Show respect for students, recognizing their diversity in terms of personal characteristics and viewpoints, expectations and academic abilities.
- Show respect for the course rationale, content and experiences.
- Devote an appropriate amount of time to working on the course, commensurate with the number of hours for which you are paid. This may involve attending additional meetings.
- Reflect on your tutoring role and practice, and take steps to improve what you do if and when appropriate.
- Pay attention to feedback you may receive on your work, and be willing to offer constructive feedback to us, to your fellow tutors, and to students.
- Mark any assessment in a prompt and unbiased fashion.

Meeting these obligations will allow you to be the best teacher that you can be.

Tutors are often asked by students ‘What is examinable?’ and ‘How do you study for this course?’ It is flippant to reply that ‘everything is examinable’ and that you ‘study by repetition’. We want students to have experience with the types of exam questions used in previous years and have therefore embedded some in the tutorials. They are a mixture of analytical, computational and descriptive questions, rather than rote learning dreary

facts. We are also running a mid-semester examination this year for practice and/or marks. The exams used to be open book, but now we only allow students to bring in two pages of notes. The students therefore have to summarize in two pages what they think is important about the course. Advise them to start compiling these notes early in the course rather than leave it to the last minute. That way, they are constantly revising and summarizing material. Make them be proactive! This is the best advice you can give them. If they simply attend classes and react to class requirements, they are not demonstrating adult independent learning and not exercising any mental discipline. We want them to be proactive and do stuff before classes (as required for SCIE1000 tutorials). If they can learn these participatory skills, then university should not be onerous.

4. Administrative arrangements

You will need to attend a meeting in the week prior to the start of classes, at which we will finalise your allocation to class times, discuss any issues with the course, and get to know the course teaching team.

Each SCIE1000 class runs for 2 hours each week, starting in week 1. You will be allocated one or more such classes. For the first such class you are allocated each week, you will be paid for 5 hours of work (comprising 2 hours of contact time and 3 hours of related work, such as preparation or marking). For each subsequent class, you will be paid for 3 hours of work (comprising 2 hours of contact time and 1 hour of related work, recognizing that you do not need to undertake preparation work for any classes after your first one each week). You will be paid for this many hours of work for each class you attend each week, irrespective of whether you complete that many hours of related work. In other words, you have the responsibility to ensure that you devote an appropriate amount of time to working as a tutor. If you are consistently spending more time on the course than you are being paid for, then you should seek to reduce that time, and are welcome to come and seek advice from us or fellow tutors. If you are consistently spending less time on the course than you are being paid for, then we may ask you to undertake some extra activities. In general, most tutors do not find that SCIE1000 tutoring requires an excessive amount of work.

You will be required to attend each of your classes, in every week of semester except one week in which philosophy tutors will take the class. You will be advised which week this is, prior to it happening. You are welcome to attend the class if you wish, but will not be paid for doing so.

SCIE1000 tutors are appointed through the School of Maths and Physics (SMP). Once we have finalized the allocations for the course, your names and details are provided to the

SMP Finance office who arranges appointment paperwork and your staff login. This can take several weeks, depending on whether or not this is your first UQ staff appointment and other factors. You may need to provide a Tax Declaration and other information – the SMP office will be in touch for this.

If you have not tutored at UQ before, you will also need to attend a Faculty of Science tutor training session, which covers all non-course-specific aspects of tutoring at UQ. This training session is held during O-week before each semester. You will be paid for your attendance.

5. SCIE1000 content

SCIE1000 is typically taken by students in their first semester of study at UQ. Thus, it plays an important role in helping students `transition' to tertiary study. Thus, we aim to make the course content interesting, engaging and challenging, and to provide a high level of support for student learning.

SCIE1000 was designed to include content from four areas:

- Mathematics, and particularly mathematical modelling.
- Science more broadly, and how mathematical models can be developed and applied in different science discipline areas. Many of the examples are from biomedical science and biology, reflecting the areas of interest of many of the enrolled students.
- Computer programming using the language Python, including developing software that implements mathematical models.
- Philosophy of science, and the nature of scientific thinking.

None of these topic areas is considered in isolation, and we continually stress the importance of "interdisciplinary thinking". Most of the mathematical content is not new to most of the students, but the applied context is. Most students have not previously learned how to program, and very few have studied any philosophy of science. We have tried hard to "weave" these four areas to form a picture that makes sense to students, whilst also challenging them to think differently about their discipline areas.

As a tutor, you will need to be comfortable with a combination of mathematics, science and computer programming in Python; other tutors will take responsibility for the philosophy component of the tutorials. If you are not broadly comfortable with mathematics, science and computer programming (at introductory levels), then you should not be tutoring this course. From the students' perspective, we want to ensure all the tutors have sufficient background to tutor this course. From your perspective, it is much more difficult for you to tutor something you are not comfortable with. Naturally,

you are paid for, and expected, to prepare material for each class. But if you cannot prepare sufficiently in about two hours then you should choose a different course to tutor.

6. SCIE1000 students

SCIE1000 is a highly recommended course for all students enrolled in the BSc, and is compulsory for all students enrolled in a Bachelor of Biomedical Science. The closely related course SCIE1100 is compulsory for all students enrolled in the Bachelor of Advanced Science (honours). Thus, students in the course have a wide range of academic abilities, and also are interested in a wide range of discipline areas. The majority of students will probably be intending to complete a major in biomedical science, and even go on to careers as doctors. However, there will be students who are majoring in mathematics, computer science, physics, biological science, chemistry, geographical science, and so on.

This means that you need to be very aware of different levels of academic ability and interest, and be able to explain the concepts to a wide range of students. For example, a student majoring in computer science may well find the Python very easy, but be completely unfamiliar with some of the biomedical examples. A student majoring in mathematics may find the mathematical content elementary, but may never have programmed before, and may not value some of the scientific contexts. Conversely, some students majoring in other areas may find the mathematics and programming quite difficult, or even non-intuitive.

You must not become impatient, frustrated, judgmental or dismissive of students who encounter difficulties in content that may be easy for you or others. If you cannot explain things to someone, perhaps one of your fellow tutors may be able to. Remember the art of teaching is being able to enlighten students, helping them to assimilate new concepts and approaches, not to show how smart you are.

7. Tutorial sheets

There are 13 tutorial sheets (one for each week of semester), although the sheet for the week of the philosophy tutorial is quite short. Each sheet typically includes:

- Readings and work that students must complete prior to coming to class (to increase the value of what happens in the class). This work is assessed.
- Some questions they need to discuss briefly with other students (to get a range of perspectives on how to approach problems)
- Some questions involving mathematics and programming, relating to the broad context of the tutorial sheet, to be completed in class.

- A question that combines mathematics and programming that students must complete for assessment
- Additional practice questions, often from previous exam papers, which students should aim to complete in class, or in their own time if they need extra examples.

You will be given an electronic copy of all tutorial sheets at the start of semester, along with solutions to each sheet. Do not distribute copies of solutions to students; we will make them available electronically the week after the classes are held. Students will also be given electronic copies of all tutorial sheets at the start of semester, and they are welcome to access these sheets on the computers in the tutorial room.

8. Tutorial classes

Tutorials are all held in a computer laboratory. Students will need to answer some questions on paper, and will need to use the Python programming language to answer other questions. Students should each sit at an individual computer, but they are welcome to work with other students around them. However, it is important that every student takes responsibility for their own learning, rather than allowing someone else (another student, or even a tutor) to do all of the work.

There are typically around 60 students enrolled in each tutorial class, and students are expected to stay for the whole 2 hours of each class. There are usually three tutors allocated to each class, and you will need to decide with your fellow tutors how the classes should operate (provided you are consistent with this document). You could designate a 'tutorial leader' if you wished, perhaps rotating this role amongst all tutors in different weeks.

Tutorials are not intended to be lectures. It is perfectly fine, even desirable, for tutors to spend some time demonstrating or explaining something to the whole class. This can be an efficient way to cover important points that are arising commonly. However, the bulk of tutorial time should be devoted to students working by themselves or with other students around them, and able to ask questions of the tutors when required.

You should structure your classes so that if a student can complete a question quickly, they can work on subsequent questions, without needing to wait for others to catch up.

9. Before classes

As part of your preparation for your tutorial class, you will need to work through tutorial questions and solutions, making sure you understand what the questions mean and how they should be answered. You should also think about how you might give hints, rather than giving away the entire answer.

You will probably want to print copies of both the tutorial sheet and the solution sheet for each of your classes, but that's up to you.

You should make contact with your fellow tutors, to decide whether any special planning or class structure is needed.

You must show up to class on time, and be ready to start helping students right away.

If you are sick or unable to attend a class for an unavoidable reason, please send an email to the list of current tutors, seeking a volunteer to take your class. If you are so sick that you cannot do this, then you need to contact one of the course coordinators. Otherwise, you are responsible for arranging a replacement.

10. Grading work and recording marks

There are multiple types of grading that you may need to undertake.

First, students need to complete some work prior to attending most tutorial classes. The tutorial sheets make it clear when this is the case. This work must be submitted at the start of the session, and cannot be undertaken in class. In the interests of efficient use of time for learning, you should collect the submitted work and mark it later. Provided students appear to have taken the tasks seriously and completed most or all of them, they should receive full marks for this component. The tutorial sheet will indicate how many marks it is worth. If students have done a poor job, then mark them down accordingly.

Second, most tutorial sheets include marks for in-class work. There is usually a particular question to be completed in class, with students demonstrating their program and/or answers to you. The tutorial sheet makes it clear which question this is, and how many marks the in-class work is worth. It is more important that students make a serious effort to do the questions that it is that they get the answers correct. Here are some marking criteria:

- If a student stays for the full tutorial session and appears to be working well and getting answers reasonably correct, they should receive full marks.
- If a student stays for the full tutorial session but either appears not to be working well or are getting things fairly wrong, they should receive half marks.
- If a student leaves early after working well and completing every question on the sheet, they should receive full marks.
- If a student has a compelling reason to leave early without having completed every question on the sheet, they should receive full marks or part marks on your judgement, based on how much work they appear to have completed.
- If a student leaves early without a compelling reason, they should receive at most half marks on your judgement, based on how much work they appear to have completed.

Third, there will almost certainly be some ad-hoc grading for you to complete during semester. You will be advised before this happens. For example, you may need to grade some programming assignments for SCIE1100 students. There will also be a midsemester exam for all students that you will need to mark. However, whatever we decide to do, there is unlikely to be a particularly heavy marking load for you.

All of the marks you allocate must be recorded on the mathematics marks system, accessible at www.maths.uq.edu.au/marks. The system is simple to use, and you should aim to remain keep relatively up to date with entering your marks (so do not leave entry until the end of semester).

11. Some final words

If you encounter any problems in your role, please contact the teaching team, including other tutors. We welcome any good advice you might have, and are also happy to give you some ideas on how to approach different situations.

We have explained the importance of SCIE1000, and your role as a tutor in the class. We also want to highlight how a tutor's job can be very satisfying. If you approach this with the right attitude, then it is fulfilling and enjoyable when you help another person understand a difficult concept, or see the beauty in assimilating new knowledge and skills. Please take the time to reflect on your role as a teacher and mentor, and, as well as providing and outstanding educational experience, also have some fun!

The SCIE1000 teaching team.