Guide to the NCSS Challenge 2013

The NCSS Challenge is a five week competition that teaches participants how to program. The Challenge is designed to give high school students an opportunity to learn and experience computer programming for themselves.

The Challenge has three different streams to cater for students of different abilities, experience and interest. Material is provided online and backed up with forum access to high quality tutorial support provided by students and staff from University of Sydney.

Quick Facts

- **Aim:** The Challenge is a fun competition in which students can learn to program and then test their skills with some fun (and challenging) questions.
- **Dates:** 5 weeks from Monday 5 August to Sunday 8 September 2013.
- **What do I get?** Students are provided with weekly notes and questions along with online access to a tutorial team for providing help and answering questions.
- **Registration:** Students can either register themselves and participate from home outside school hours or schools can register students as a group and support participation from school as a classroom activity (once registered, students have 24-hour access to the site from any Internet connected computer).
- **Difficulty:** The competition starts at a quite easy level in Week 1, and moves through to more challenging questions by Week 5. Although the hardest questions are tough enough to test the best programmers, we make sure there are some questions each week that everyone can do.
- **Levels:** There are three streams catering to students of varying ability and interest: Beginners, Intermediate and Advanced.
- **Eligibility:** Any school student can enter (see the information on each stream in this document to help you choose the stream that is right for you or your students).
- **Cost:** The cost is $20 (which includes GST) per student per stream. Students can also be enrolled through their school to an annual subscription to all streams, costing $30.
- **Teachers:** As a teacher, you must enrol and pay for the NCSS Challenge streams you want access to. You may purchase an annual subscription to all courses at a discounted price of $30.
- **Register online** at challenge.ncss.edu.au
- **Please note:** Once you have registered it is not possible for us to offer you a refund, so please read this brochure carefully and contact us if you have any questions before registration.

More Information

What do I get when I register?

Each student or teacher who registers for the Challenge gets:

- The opportunity to participate in the 5-week NCSS Challenge, with a set of 5-10 fun questions each week.
- A full set of notes (downloadable as a PDF) for each week of the Challenge, that you can use to study in your own time.
- Fully worked solutions and commentary, released at the end of each week.
- Online tutorial support (via online forums) from highly qualified students and staff from University of Sydney

If you participate over the full 5 weeks of the Challenge you will receive a certificate from The University of Sydney showing your level of achievement or participation.

We encourage teachers to use the notes and questions as a learning exercise in their classroom, and are happy to provide more information about this.

How does the Challenge Run?

The notes and Challenge questions for the week are released at 9am each Monday morning for the five weeks of the Challenge. Students have until midnight the following Sunday to submit their answers to collect marks. Answers can be submitted later but, as the solutions will then be online, no marks can be allocated for late submissions. Marks are not allocated in the first week to give students a chance to gain familiarity with the system.

To answer a Challenge question the student needs to write a computer program and ensure it works correctly. Once satisfied they submit the source code to their program to the online marking system which runs a series of tests to determine if the program is correct. If the program fails testing then feedback is provided as to why it failed and the student has the option of re-submitting. Multiple attempts to each question are allowed but after 5 attempts the amount of marks allocated to a finally successful submission diminishes.

Students in the *Beginners*, *Intermediate* and *Advanced* streams are shown how to download and install Python on their own computer. Students then are able to write, run and debug Python programs locally before submitting the source code for marking.

The Challenge web site hosts online forums for the students to ask questions and communicate with others doing the Challenge. As some students may feel intimidated by open forums the site also has closed forums where students can address questions directly to our experienced team of tutors.

Support is provided for students to answer questions each week to reinforce learning. Support can be of the form of answering general questions on the programming concepts or providing feedback on source code submitted to the marking system. The *Advanced* stream will emphasis the competitive aspect of the Challenge by providing minimal support for a small number of questions each week. In all cases students are encouraged in the first instance to work out solutions themselves and help is in the form of guidance to see how a problem may be solved.

The marks of the top students for each of the three age groups of students (Years 7-8, 9-10 and 11-12) are displayed on the web site for all participants of that course to see.

Registration and refund policy

Note that once you have registered it is not possible for us to offer you a refund, so please read this brochure carefully and contact us if you have any questions before registration.

What level should I enrol in?

In 2013 there are 3 different streams available in the Challenge. We have included a brief description and examples from each, so you can judge what level is most suitable for you. After the first week of the Challenge you can change streams if the one you enrolled in isn't suitable.

A small number of students (mainly those who have enjoyed the Challenge previously) choose to enrol in multiple streams. For some this will be a large workload and we generally encourage students to attempt just one stream at a time. In any event the no refund policy will apply.

Year 12 students may enrol but we realise that the timing of the Challenge may conflict with preparation, or participation, in end of school exams. Some Year 12 students choose to enrol but do not complete the full five weeks. If in doubt we suggest students concentrate on their exams in Year 12 and, ideally, participate in the Challenge in earlier years.

We are happy to enrol teachers in previous years of the Challenge free of charge so they can gauge first hand the level and pace of the streams and get a better idea of the topics covered. Please email us to organise this access.

Stream 1: *Beginners*

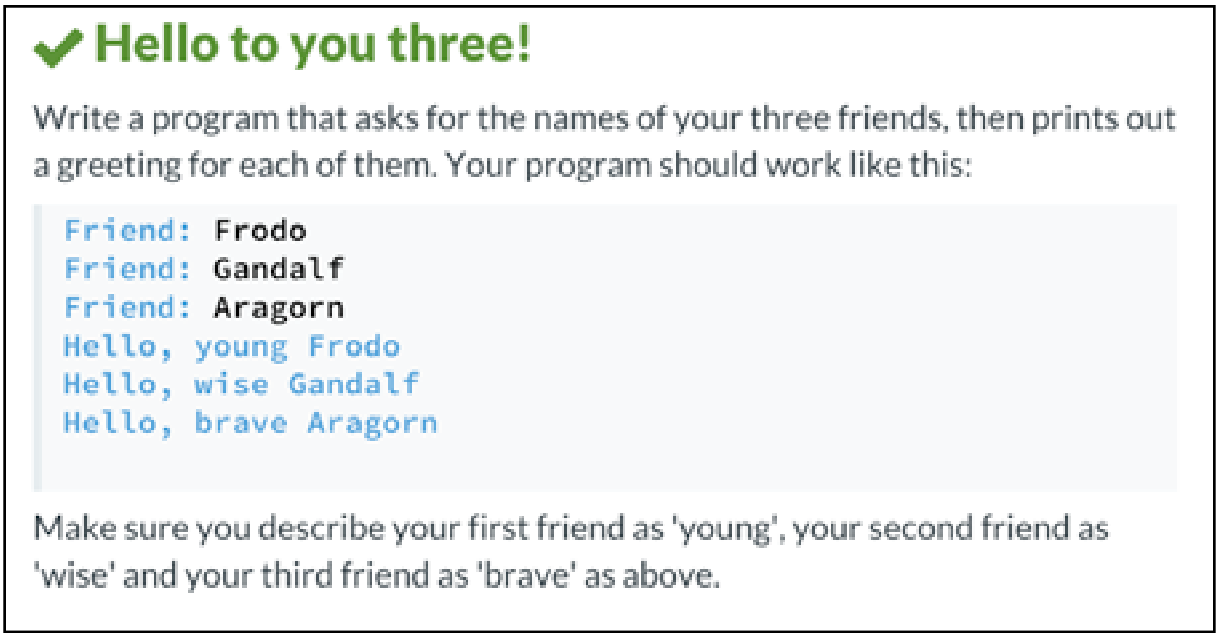
The *Beginners* stream is designed for students with no prior experience at programming. It starts at the very beginning, and includes extra questions each week that cover the basic concepts.

This level is most suited to students in years 7 to 10.

Topics Covered:

- The concept of programming
- Variables and user input and output
- Python data structures for strings and lists
- String manipulation
- Python control structures: for loops, while loops, if statements

Sample Question: Week 1 *Beginners*



✓ Hello to you three!

Write a program that asks for the names of your three friends, then prints out a greeting for each of them. Your program should work like this:

```
Friend: Frodo
Friend: Gandalf
Friend: Aragorn
Hello, young Frodo
Hello, wise Gandalf
Hello, brave Aragorn
```

Make sure you describe your first friend as 'young', your second friend as 'wise' and your third friend as 'brave' as above.

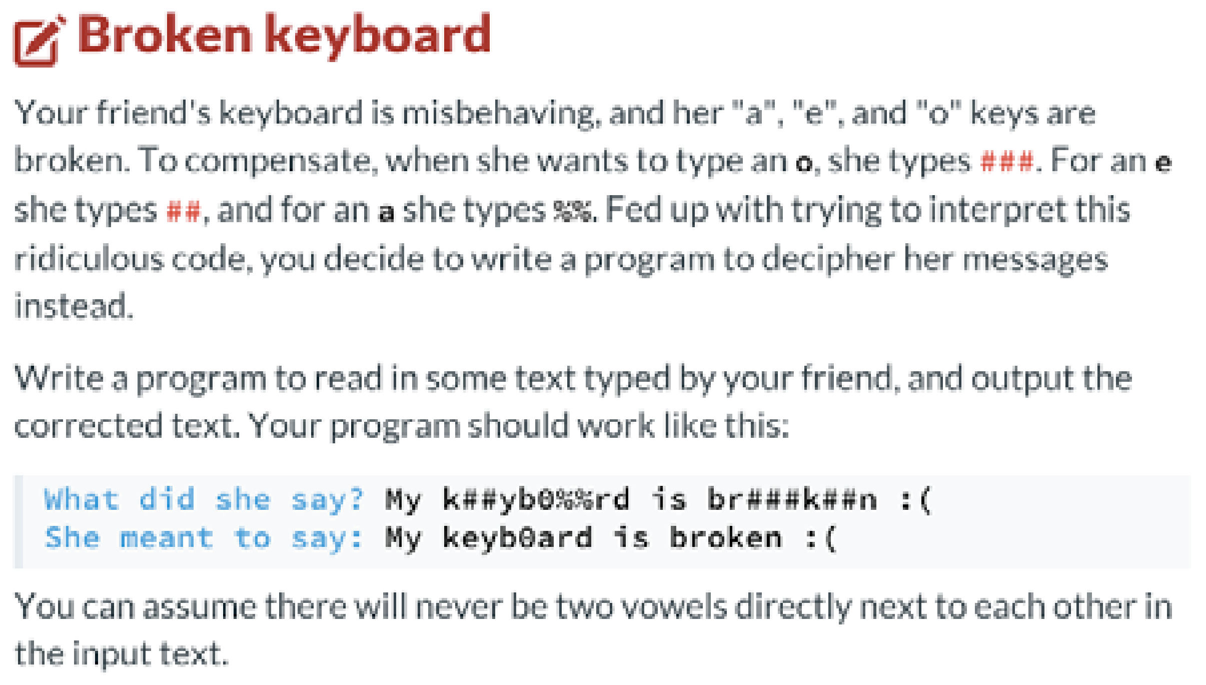
Stream 2: *Intermediate*

The *Intermediate* stream is designed for students who either have completed the *Beginners* stream in a previous year, have some other prior experience at programming, or who are naturally quite capable students who can start with material more advanced than the *Beginners* stream. Advanced maths and science students in years 10 and above may also want to enter this level. If you are enrolling a class you may find you have one or two students who would benefit from starting here.

Topics Covered:

- The concept of programming
- Variables and user input
- String manipulation
- Python control structures: for loops, while loops, if statements
- Python data structures: lists, dictionaries
- Functions

Sample Question: Week 1 *Intermediate*



Broken keyboard

Your friend's keyboard is misbehaving, and her "a", "e", and "o" keys are broken. To compensate, when she wants to type an **o**, she types `###`. For an **e** she types `##`, and for an **a** she types `%%`. Fed up with trying to interpret this ridiculous code, you decide to write a program to decipher her messages instead.

Write a program to read in some text typed by your friend, and output the corrected text. Your program should work like this:

```
What did she say? My k##yb0%%rd is br###k##n :(
She meant to say: My keyb0ard is broken :(
```

You can assume there will never be two vowels directly next to each other in the input text.

Stream 3: *Advanced*

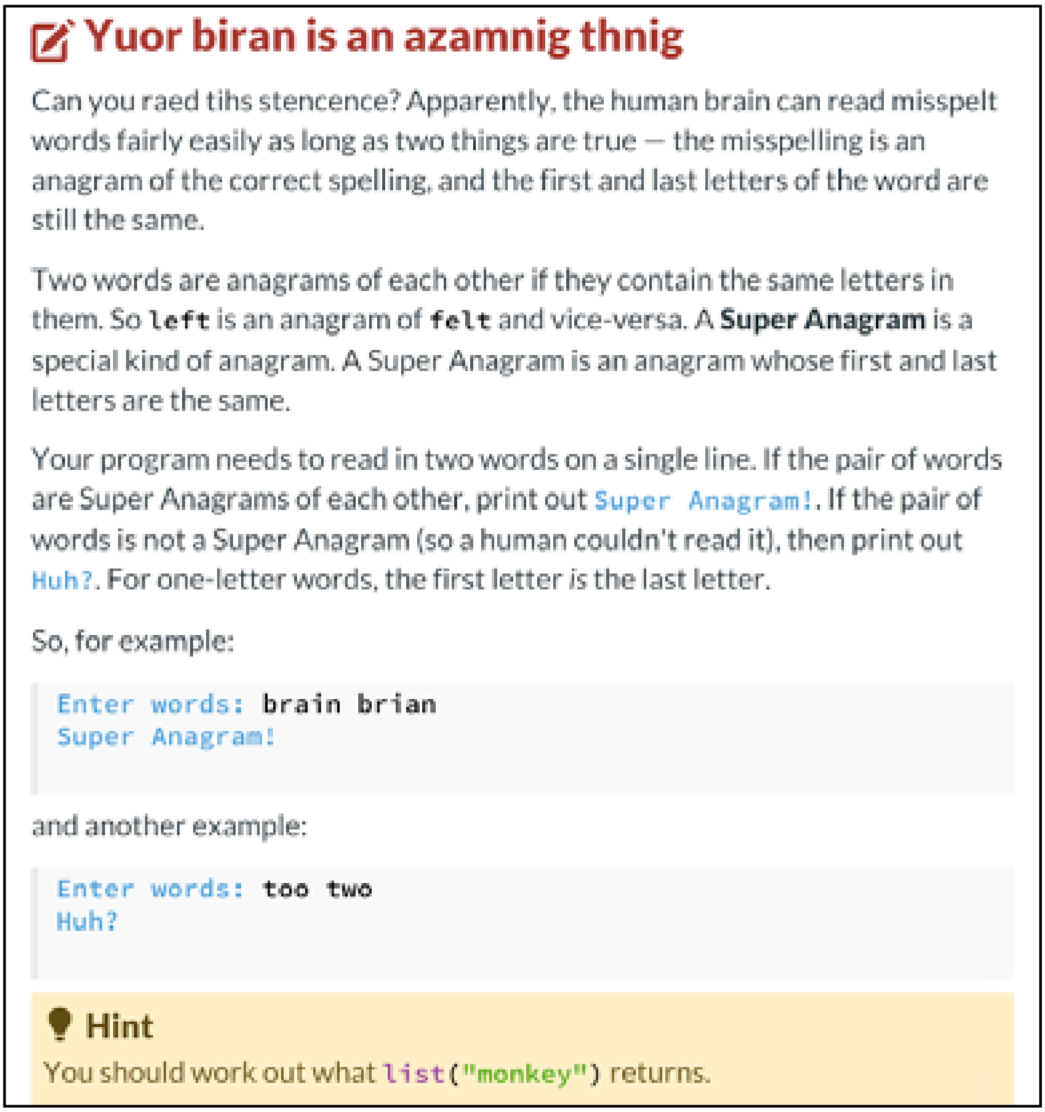
The *Advanced* stream is designed for students who have completed the *Intermediate* stream in a previous year and want more advanced material or who have significant programming experience. While some capable students can start the Challenge at the *Intermediate* stream level those students who are new to the Challenge should not take this as their first course unless they really have significant programming experience.

This level is most suited to students in years 10 to 12, or extremely talented junior students.

Topics Covered:

- All topics listed in the *Beginners* and *Intermediate* streams, plus:
- Advanced functions and recursion
- Reading from and writing to files
- Regular expressions
- Object oriented programming
- Exception handling

Sample Question: Week 1 *Advanced*



Yuor biran is an azamnig thnig

Can you raed tihs stencence? Apparently, the human brain can read misspelt words fairly easily as long as two things are true — the misspelling is an anagram of the correct spelling, and the first and last letters of the word are still the same.

Two words are anagrams of each other if they contain the same letters in them. So `left` is an anagram of `felt` and vice-versa. A **Super Anagram** is a special kind of anagram. A Super Anagram is an anagram whose first and last letters are the same.

Your program needs to read in two words on a single line. If the pair of words are Super Anagrams of each other, print out `Super Anagram!`. If the pair of words is not a Super Anagram (so a human couldn't read it), then print out `Huh?`. For one-letter words, the first letter *is* the last letter.

So, for example:

```
Enter words: brain brian
Super Anagram!
```

and another example:

```
Enter words: too two
Huh?
```

Hint

You should work out what `list("monkey")` returns.

Our sponsors

NCSS is brought to you by the University of Sydney with generous support from our sponsors.

Gold Sponsors

- CSIRO
- Freelancer
- Google
- NICTA
- NSW Trade and Investment
- Optus
- WiseTech Global

Silver Sponsors

- Atlassian
- Defense Signals Directorate

Bronze Sponsors

- ResMed

Contact

For **administrative** queries, please email
E: info@ncss.edu.au

For **payment** queries, please email
E: payment@ncss.edu.au

For **technical** queries, please email
E: challenge@ncss.edu.au

For all other queries, and information about **other programs** we run, including the **NCSS Summer School**, please contact
Nicky Ringland
School of IT, J12
University of Sydney NSW 2006
AUSTRALIA
T: +61 2 9036 9712
E: ncss@ncss.edu.au
W: www.ncss.edu.au

Brought to you by:

Gold sponsors: