SCIE1000 Theory and Practice in Science
Python and Communication Assignment
Semester One, 2020

1 The scenario

A public science museum in St Lucia is planning to update its exhibit. A feature of the museum is that each exhibit item is accompanied by two explanations, each written for a different audience. One explanation is pitched to the "science rookie" and the other to the "science enthusiast". Patrons read the explanation tailored to the level at which they feel most comfortable. Some characteristics of a typical audience member in each category are described in Table 1.

Table 1: Characteristics of different patrons

Patron Type	**Typical characteristics**
Science Rookie	Not familiar with scientific terminology or notation; will need terminology explained using a simple vocabulary; is unfamiliar with graphs; may be a younger person, possibly 10+ years of age; likes to press buttons.
Science Enthusiast	Familiar with common scientific terminology and notation (not overly technical); will need terminology explained using somewhat sophisticated vocabulary; is prepared to read longer passages of moderate complexity; is familiar with graphs; likes to press buttons.

The planned exhibition is called "Exploring Our Galaxy". The topic is exoplanets (planets found orbiting distant stars) and the aim is summarised in the following exhibition prospectus:

> *With this exhibition we aim to instil in our patrons a sense of wonder at the vastness of our galaxy and the potential for finding other forms of life with whom it may be possible to communicate. Patrons will marvel at the challenge of searching for exoplanets and they will speculate about the number of potential civilisations in our galaxy with whom we could theoretically communicate.*

The museum director has asked the SCIE1000 teaching team for help in finding skilled volunteers to develop exhibit items. Once developed, the items will be maintained and potentially modified by museum staff, each of whom has a strong background in high-school mathematics, together with at least a beginners level of Python experience. We assured the director that SCIE1000 students are skilled at: making mathematical models using a mathematical toolkit familiar to any student of Maths B (or equivalent); writing Python programs, including those which use arrays, loops, plots and new functions; and communicating scientific information to various audiences.

Based on this boasting by the SCIE1000 teaching team, you have been asked to develop an exhibit item. You will develop an interactive (command-line) Python program which engenders in museum patrons a sense of wonder at the vastness of our galaxy and the potential for communicating with other forms of life.

2 An overview of the task

You will write an interactive Python program to guide a user through some speculation about civilisations in our galaxy (see Section 4) as well as whether a potential exoplanet could be detected from Earth given its size and distance from its star (see Section 5).

Your program will **follow the logical flow laid out in the flow chart provided in Figure 2**. A detailed list of program requirements is provided in Section 6 of this document.

Your assignment will be separately be graded on two aspects, each on a 1–7 scale. The first aspect will be your use of Python to represent the underlying mathematical models. This will include the quality of your code and the accuracy with which you represent the models. The second aspect will be on the communication that you use. This covers both the communication within your program (for staff at the exhibit) and the communication you use with the patrons of the exhibit.

Your submitted code will be run and tested as part of this grading process. A rubric (marking criteria) for this assignment is on the last page of this document. This assignment has an **advanced section** which must be attempted by students aiming for grades of 6 or 7 (see the grading criteria for more explanation). The shaded section of the flow chart indicates this advanced section, and the corresponding modelling information is in Section 5.6. If you have any questions, please contact the course staff.

Your code file must be uploaded via the Blackboard submission link by **2pm on 1 June, 2020**. Late submissions without an approved extension will be penalised; consult Section 5.3 of the Electronic Course Profile for more information concerning late submissions.

3 About getting help

This assignment is a piece of summative assessment, designed to let you demonstrate your level of mastery of several learning objectives in this course. As such, **it is very important that the work you submit is all your own**.

This does not mean that you cannot receive help in regards to this assignment, but that help must be limited to general advice about Python and modelling - not specific to how to do this particular assignment. Your teaching team, the SLC tutors, your classmates, your friends, and anyone else for that matter, can answer as many general questions about Python and modelling as you care to ask. They can even help you understand what particular error messages may mean. They should not, however, tell you what to write or correct your code. You should type or create every character in the files you submit. **The files that you submit will be checked using software which is specially designed to detect plagiarism in code.** Consult Section 6.1 of the Electronic Course Profile for more information and procedures concerning plagiarism.

This task sheet has been carefully constructed, and part of your job is to interpret the information it contains. **Some choices have been left to your judgement, and this is intentional.** Any questions you have about the assignment task should be posted on the course Piazza site as a public post (visible to all students). This is the only place where you can receive authoritative answers to questions. In this way, all students will have access to the same information. Sometimes the answer to a question on Piazza will be "See the assignment task sheet". Such answers are simply to avoid restating information, and indicate that you will need to decide how to use the supplied information.

4 Imagining potential civilisations in the Milky Way

The **Drake equation**, named after the astronomer and astrophysicist Dr. Frank Drake, is a formalisation to guide speculation about the probability of finding civilisations in our galaxy (the Milky Way) with whom it may be possible to communicate. This equation is often quoted with seven parameters (see [1]). We will use the following simplified version:

$$N = R \cdot p \cdot n \cdot c \cdot L \tag{1}$$

where

N = number of civilisations in the galaxy that can communicate with Earth
R = average rate of star formation (per year) in the galaxy
p = proportion of stars with planetary systems
n = number of planets per system with conditions suitable for life
c = proportion of potentially habitable planets on which a technological civilisation develops
L = average lifetime (in years) of such a civilisation within the detection window

Table 2, adapted from [1], includes recent estimates (ie "best known estimates") for the parameters in the Drake equation, as well as historical estimates which were used in the 1960's.

Table 2: Parameters of the Drake equation and their estimates

Parameter	R	p	n	c	L
1960's estimation	10 per year	0.5	2	0.0001	10,000 years
Recent estimation	7 per year	0.5	1	0.02	10,000 years

Before you plan your communication to the user about this information (see the flow chart in Figure 2 for details) you should carefully think about how estimates for N are made and how reliable you think those estimates are.

5 Exoplanets

5.1 Detecting exoplanets using the Kepler space telescope

Extraterrestrial planets, or exoplanets, are planets that orbit around stars other than our Sun. Since the nearest star is around 4 light-years away (the distance light travels in 4 years), exoplanets are extremely difficult to detect. However, in recent years, a number of different techniques have been developed which are capable of directly or indirectly inferring the existence of an exoplanet around another star in our galaxy (the Milky Way).

One very successful method for detecting exoplanets is to observe the intensity of the light emitted by the star as a function of time. If an exoplanet passes in front of this star, it partially blocks the star as viewed from the Earth, and the measured intensity of the star will (slightly) decrease. Multiple measurements at regular intervals (at least three passes in front of its star) can be used to confirm the existence of an exoplanet.

A telescope that has been used extensively for the detection of exoplanets using this approach is the Kepler space telescope. Its goal was to detect smaller exoplanets in the range from the size of Earth to the size of Jupiter. This telescope, recently retired, was launched in 2009, and has successfully detected several thousand exoplanets [2].

5.2 Modelling exoplanet detection

The model we will develop here will be a very simplified model of the physical process in which a planet transits in front of a star. In particular, we will make the following assumptions when building our model:

- We will be detecting exoplanets that are orbiting a star which has the same size and mass as the Sun. The star mass impacts on the speed of the exoplanet as it moves around its star, and hence the time to complete a full orbit around the star, while the mass and size of the star influence the time the exoplanet takes to pass in front of the star.
- There is a "perfect" alignment of the exoplanet between our observation point on Earth, and the star around which it orbits. This maximises the time that the exoplanet is in front of its star.
- The light emitted by the star is uniform across the width of the star. This simplifies our calculation of intensity as the exoplanet transits in front of the star.
- The radius of the exoplanet is small compared to the radius of its star. This allows us to choose a simple model of the transit.

Note: there will be some constants relevant to our solar system that you will need to research and find values for. Exercise care with units!

5.3 Choosing input parameters

To model the transit of the exoplanet in front of its star, we first need to specify the the size of the exoplanet and the distance the exoplanet is from its star. Your program will ask the patron to choose these. However, most rookies (and many enthusiasts) will not have a good feel for what values would be appropriate. Hence, you will need to think carefully about how this is posed to the patrons. We recommend thinking about how to use relative values compared to similar values for the Earth/Sun.

5.4 Model - calculating the velocity of the exoplanet

The velocity of the exoplanet is dependent on how far it is from its star, and the gravitational attraction of the star. We can relate these back to values for Earth using

$$\frac{\text{velocity of exoplanet}}{\text{velocity of Earth}} = \sqrt{\frac{\text{distance of Earth from the Sun}}{\text{distance of exoplanet from its star}}}$$

5.5 Model - calculating the key output parameters

We can now begin calculating the relevant output parameters. These are:

- **Minimum relative intensity.** When the exoplanet is fully between the Earth and the exoplanet's star, the intensity of the star observed from the Earth will be decreased as the exoplanet blocks some of the light from the star. We can define a relative intensity as the ratio of the observed intensity when the exoplanet is in front of the star to the observed intensity when the exoplanet is not in front of the star. This intensity depends on the ratio of the cross sectional area of the exoplanet to the cross-sectional area of its star. The cross-sectional area of a sphere is the area of a circle with a radius which is equal to the radius of the sphere. We can thus write an equation for the observed relative intensity of the star when the exoplanet is in front of the star as

$$\text{Relative intensity} = 1 - \frac{\text{cross-sectional area of the planet}}{\text{cross-sectional area of the star}}$$

  This equation gives the minimum observed relative intensity. The maximum relative intensity, when the exoplanet is not in front of the star, is equal to 1.

  Note that we have ignored the short period of time when the exoplanet is only partially in front of the star (i.e. across one edge of the star). We will explore this further in the advanced section below.

- **Transit time.** The velocity of the exoplanet and the diameter of its star will determine the transit time. The faster the exoplanet is moving, the shorter the transit time and, likewise, the larger the diameter of the star, the longer the transit time. The transit time can be written as

$$\text{Transit time} = \frac{\text{diameter of the star}}{\text{velocity of the exoplanet}}$$

- **Period of orbit.** The period of the orbit of the exoplanet is the time for the exoplanet to make one complete orbit around its star (this is 1 year for Earth). The period of orbit is determined by the exoplanet's speed, and the distance the exoplanet is from its star. The period can thus be written as

$$\text{Period} = \frac{\text{circumference of the orbit}}{\text{velocity of the exoplanet}}$$

Your program should output values for these parameters, accompanied with appropriate explanations for what each means.

- **Detection.** The detection limit for Kepler to observe a planet is an intensity decrease of 1 part in 10,000 (or 0.01%) as the exoplanet transits the star. You should inform the patron whether their chosen exoplanet could be detected or not. You might also like to comment on how long the star needs to be observed to confirm detection of the exoplanet.

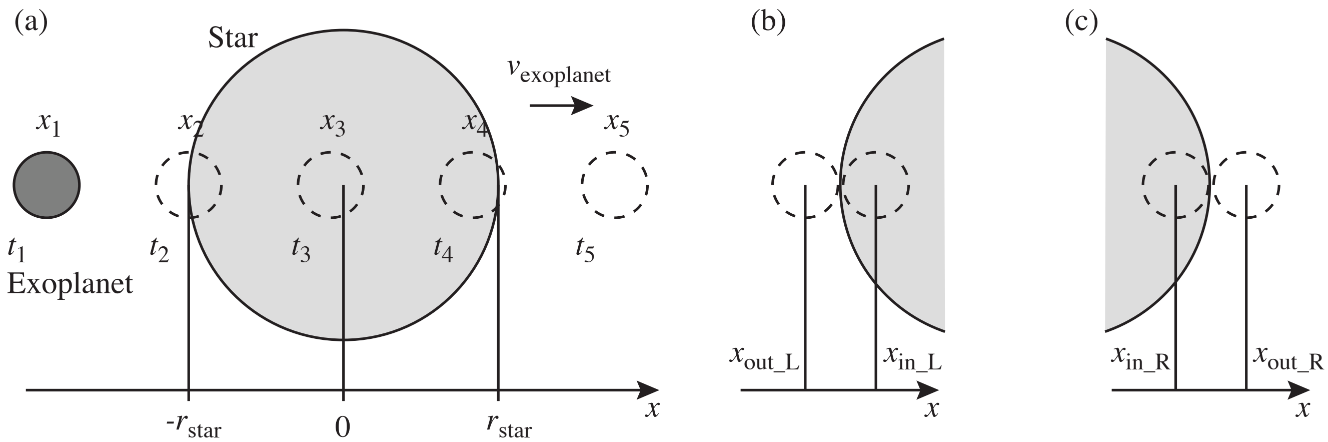


Figure 1: For the advanced section. Diagram of the motion of the exoplanet across the face of the star. (a) Stepping the position of the exoplanet (x) as the time is varied (t); (b) Limits for the exoplanet crossing the border of the star on the left side; (c) and limits for the right side.

5.6 Advanced section

The science enthusiast should be provided with further information including a graph of the intensity and a paragraph of accompanying text. The graph should show how the relative intensity varies as the exoplanet transits its star over time. The accompanying text should briefly (but clearly) explain one limitation of the model that has been used.

To create the graph, there are a number of approaches that you could use

- Develop a time step approach. This will involve using a loop in your program which steps through time as the planet transits in front of the star. This is shown in Fig. 1(a), with a very coarse time step. At each time step, you will need to calculate the position of the planet. Time, t and position, x, are related via

$$x = x_0 + vt$$

  where x_0 is some starting position ($t = 0$), and v is the velocity of the planet. At each time step, you will need to calculate the observed relative intensity which will be

  - equal to 1 if the planet is not in front of the star (positions 1 and 5 in the figure).
  - equal to the minimum intensity, I_{min}, that you previously calculated if the planet is fully in front of the star (position 3 in the figure).
  - between the minimum intensity and 1 when the planet overlaps the edge of the star (positions 2 and 4 in the figure). We will use a linear interpolation to approximate the intensity

$$\text{Relative intensity} = 1 - \frac{x - x_{\text{out}}}{x_{\text{in}} - x_{\text{out}}} \times (1 - I_{\text{min}})$$

    where x is the position of the planet, and x_{in} and x_{out} are the positions of the planet when it is just inside and just outside the star, respectively (see Fig. 1b and Fig. 1c).

- Determine the times of specific events, and join using straight lines. As noted in the previous approach, the intensity if equal to 1 when the planet is outside the star, and equal to $I_{\min}$ when the planet is fully in front of the star. The transition between these intensities occurs as the planet crosses the border of the star. In this approach, you will need to determine the time (relative to some starting time) when the planet is just inside the star, and just outside the star, on each side of the star. These positions are shown in Fig. 1(b) and Fig. 1(c).

If you are unsure of what your graph should look like, then we recommend that you do some research on the Kepler mission.

Some extra notes on Python graphing:

- Python sometimes plots a graph using an offset. For example, for an axis with limits of 0.999 - 1.000, Python may create a graph with the axis running from 0 - 0.001 with an offset of +0.999. To avoid this use the Python command: `ticklabel_format(useOffset=False)`

- If you want to specify the horizontal axis limits, use the command `xlim(lowerlimit,upperlimit)` and if you want to specify the vertical axis limits, use the command `ylim(lowerlimit,upperlimit)`, where you choose lowerlimit and upperlimit yourself (put some numbers in to see how it works).

6 Specifications for your submitted file

The file you submit for this assignment must be an interactive Python program which models certain aspects of the search for exoplanets and other civilisations in our galaxy.

Specifications about the Python:

- Museum staff have supplied a **flowchart** describing how the program should run (Figure 2). Your code must be an implementation of the flowchart provided.

- Your code must be well-structured and follow the guidelines for programming practice, as introduced in SCIE1000.

- Whenever you prompt the user for information, you may assume they enter a number, and you can store their answer as a **float**.

- You may only use Python commands **introduced in SCIE1000**. Recall that museum staff must be able to maintain and modify the code, so you may only use commands that they understand. Museum staff have a beginner's level of experience using Python, which you may regard as the equivalent of a student who has taken SCIE1000. The Python commands you have covered in this course should be more than sufficient to complete the assignment.

- Museum staff have identified several functions that they think will be useful in possible modifications and extensions of the code. You **must define these functions in your code**, with the exact names specified below and which take the same arguments in the order specified. You should call these functions in your code as appropriate. You may define other new functions as needed.

(a) You must define a function called `get_period_of_planet` which takes two arguments, the distance from the exoplanet to its star and the velocity of the exoplanet (in that order), and returns the period of the orbit of that exoplanet around its star.

(b) You must define a function called `get_transit_time` which takes two arguments, the velocity of the exoplanet and the radius of the star (in that order), and returns the time taken for the exoplanet to transit across the front of its star.

(c) You must define a function called `get_min_rel_intensity` which takes two arguments, the radius of the exoplanet and the radius of the star (in that order), and returns the minimum relative intensity of the light from that star during the exoplanet's transit across the star.

Specifications about the communication:

- **All** messages to the user, including prompts to enter data, should communicate in a manner **appropriate for the level of patron** and should serve the purpose of the program.
- You should write no more than a couple of sentences for each piece of information you explain to the user. Follow the principles for communication in science as described in Appendix B of the lecture book. Be precise, clear and concise!
- You should **use units appropriately** in your communication with the user. Make sure you are aware of the units of values being passed into functions and the units of values being returned from functions.
- You should include useful and appropriate comments in your code to help the museum staff who may need to maintain and modify the code. Any variable names and function names you define should be chosen with communication in mind.
- Whenever you produce a graph you should provide appropriate labels and explanatory text.

File type and file name:

- Your assignment should be saved as a .py file. The file should be called

  InteractiveSpaceAliens********.py

  with the string ******** replaced by your student number.
- When you are writing one long program as a .py file, it is usually easiest using Spyder, rather than Jupyter. To access Spyder, simply open Anaconda and then click "Launch" under the option for Spyder.

References

[1] Glade, N., Ballet, P. and Bastien, O. (2012) A stochastic process approach of the drake equation parameters. *International Journal of Astrobiology*, Vol. 11(2), pp. 103–108.

[2] Cleary, D. (2018). Planet hunter nears its end: Kepler space telescope found trove of exoplanets. *Science*, Oct 19, 2018, Vol. 362(6412), p. 274(2).

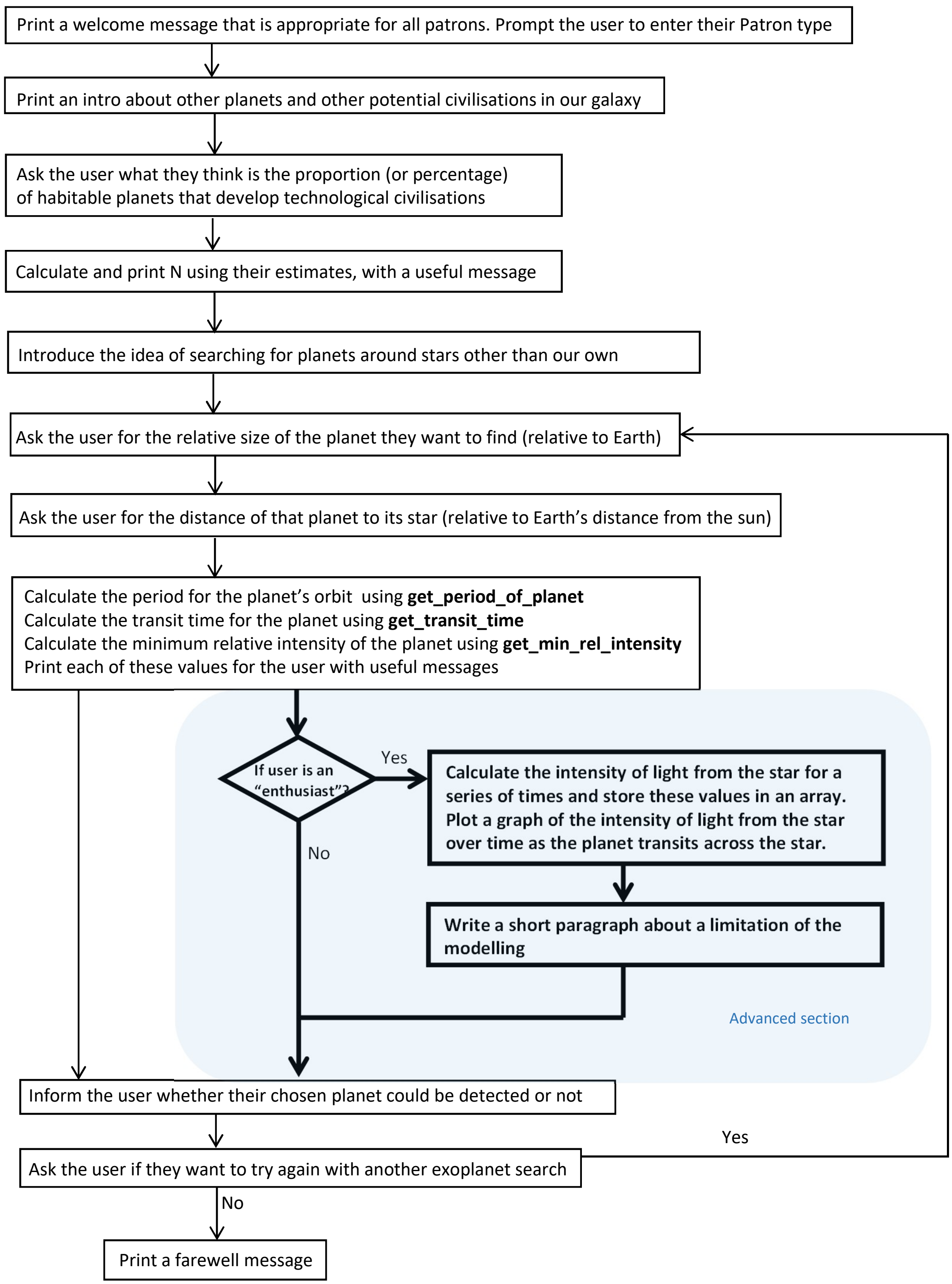


Figure 2: Flowchart for the interactive program (shaded section indicates the advanced section).

Assignment Grading

Your grades for the Python and Communication sections of the assignment (each on a 1–7 scale) are calculated by using the grade that best matches your answers for the main sections plus the extra grades for the advanced sections. Your overall grade will thus be up to a maximum of 14. The table below shows the criteria for each grade.

Grade	**Python (1–7)**	**Communication (1–7)**
1	The code is limited and displays little understanding of the modelling involved.	Communication is very poor and would be difficult to understand by any audience.
2	The code fails to run for any input, does not meet the specifications, and/or has serious conceptual errors in the modelling.	The communication of the relevant scientific information (in the code and to patrons) is generally poor.
3	The code runs for some expected inputs, meets at least some of the specifications, and/or may have some significant conceptual errors in the modelling.	The communication of the relevant scientific information (in the code and to patrons) lacks clarity or is not at the appropriate level for most target audiences.
4	The code runs without error for most inputs, mostly meets the specifications, and mostly represents the mathematical modelling with perhaps only some conceptual and/or mathematical errors.	The communication of the relevant scientific information (in the code and to patrons) is adequate for most target audiences. The may be some limitations in the communication within the code and/or to patrons.
5	The code runs without error for all expected inputs, largely meets all the specifications, and accurately represents the modelling with perhaps only one or two minor errors.	The communication of the relevant scientific information (both within the code and to patrons) is mostly clear, fluent, and uses a level and style appropriate for each target audience.
Advanced Section		
0	Not completed, or the advanced code has significant conceptual errors.	Not completed, or the communication in the advanced section is poor.
+1	The code for the advanced section runs with only minor errors and mostly represents the modelling with some conceptual or calculation errors.	The level of communication is adequate throughout the advanced section but would require some editing before use in an exhibit.
+2	The code for the advanced section runs without error and accurately represents the modelling with at most one minor error.	The level of communication is clear throughout the advanced section and would be suitable for use in an exhibit with perhaps only one or two small changes.

SCIE1100 Semester 1, 2020: Python and Communication Assignment

1 A scenario

A newly created public science museum is to open in St Lucia. A feature of the museum is that each exhibit item is accompanied by three explanations, each written for a different audience. One explanation is pitched to the "rookie scientist," another to the "seasoned scientist", and a third to the "grizzled scientist." Patrons read the explanation tailored to the level at which they feel most comfortable. Some characteristics of a typical audience member in each category are described in Figure 1.

Category of Scientist	Typical characteristics
Rookie	Usually undertaking primary or early secondary schooling; easily distracted; will not read more than a few simple sentences at a time; will need terminology explained using a simple vocabulary; unfamiliar with graphs; likes to press buttons.
Seasoned	Usually undertaking or recently completed secondary schooling; will read sentences of moderate complexity; will need terminology explained using a somewhat sophisticated, but not technical, vocabulary; familiar with graphs; likes to press buttons.
Grizzled	Usually has completed secondary schooling, perhaps quite some time ago, and some specialised science training; prepared to read longer passages of moderate complexity; will understand common scientific terminology; familiar with graphs; likes to press buttons.

Figure 1: Characteristics of a typical patron in each category

The museum is planning an exhibition called "Mother Nature, the Mother of Invention." The topic is biomimicry, and the aim is summarized in the following passage from the exhibition prospectus:

> With this exhibition we aim to engender in our patrons a sense of wonder at the super-human abilities displayed by animals, and to demonstrate how humans can acquire super-human powers through technology which copies from nature. Patrons will marvel at the complexity involved in the "computations" that animals perform intuitively, and the human cleverness involved in developing technologies which mimic natural abilities.

One of the super-human abilities to feature in the exhibition is *echolocation*, and the featured technology through which humans have acquired this power is called *active sonar*.

The museum director has asked the SCIE1100 teaching team for help in finding skilled volunteers to develop exhibit items. Once developed, the items will be maintained and potentially modified by museum staff, each of whom has a strong background in high-school mathematics, together with at least a beginners level of Python experience. We assured the director that SCIE1100 students are skilled at: making mathematical models using a mathematical toolkit familiar to any student of Maths B (or equivalent); writing Python programs, including those which use arrays, loops, plots and new functions; and communicating scientific information to various audiences.

Based on this boasting by the SCIE1100 teaching team, you have been asked to develop an exhibit item. You will develop an interactive (command-line) Python program which engenders in museum patrons a sense of wonder at the Sperm Whale's ability to judge the distance to an object in the ocean using echolocation.

2 An overview of the task

In this assignment you will construct a model of echolocation from three other models, one which predicts the speed of sound in seawater given certain parameters, another which predicts the salinity of the ocean at different latitudes, and another which predicts the temperature of the ocean in late March at different depths and latitudes. These models are described in Section 4 of this document. You will write an interactive Python program to perform calculations concerning the model of echolocation. Your program will extend the template provided in Section 5, and follow the logical flow laid out in the flow chart provided in Section 5. You will document your code with comments so that it can be easily maintained and extended by anyone with a strong background in high-school mathematics and a beginner's level of Python experience. You will provide sample output produced by running your code so that museum staff know what to expect when it runs and can believe that you have delivered code that meets the specifications.

In summary, this assignment requires you to produce and deliver two items:

(D1) A file containing a well-documented interactive Python program. A detailed list of program requirements is provided in Section 5 of this document.

(D2) A file containing sample output generated by running your program. A detailed description of what is required is provided in Section 6 of this document.

A rubric (marking criteria) for this assignment will be posted on blackboard.

All deliverables are to be uploaded through Blackboard by 2 pm on 1 June, 2020. The time-stamp from Blackboard will be used to determine the time of submission. Late submissions will be penalised. Consult Section 5.3 of the Electronic Course Profile for more information concerning late submissions.

3 About getting help

This assignment is a piece of summative assessment. It is designed to let you demonstrate your level of mastery of several learning objectives in this course. As such, it is very important that the work you submit is all your own. This does not mean that you cannot receive help in regards to this assignment, but that help must be limited. There will be no workshop time allocated to working on your assignment. You can use MyPyTutor, available through Blackboard, to practice and learn Python concepts. Your teaching team, the SLC tutors, your classmates, your friends, and anyone else for that matter, can answer as many general questions about Python and modelling in general as you care to ask. They can even help you understand what particular error messages may mean. They should not, however, tell you what to write or correct your code. **You should not look at anyone else's code for ideas, and you should not show your code to any of your classmates. Both of these actions are examples of behaviour that may be considered academic misconduct.** You should type or create every character in the files you submit.

Your teaching team and tutors will not answer questions about the nature of the task, or elaborate on the requirements, in person. Any questions you have about the assignment task should be posted on the course Piazza site. This is the only place where you can receive authoritative answers to questions. In this way, all students will have access to the same information. Sometimes the answer to a question on Piazza will be "See the assignment task sheet." Such an answer is not intended to be grumpy, but to avoid restating information. Trying to say the same thing in different places can lead to misunderstandings or unintended inconsistency. This task sheet has been carefully constructed, and part of your job is to interpret the information it contains. Some choices have been left to your judgement, and this is intentional.

The files that you submit will be checked using software that detects plagiarism. Consult Section 6.1 of the Electronic Course Profile for more information and procedures concerning plagiarism.

4 A model of echolocation

4.1 Sperm Whales

According to the Australian Government's Department of Environment and Energy [6]:

> Toothed whales (including dolphins) have developed a remarkable sensory ability used for locating food and for navigation underwater called echolocation. Toothed whales produce a variety of sounds by moving air between air-spaces or sinuses in the head. Sounds are reflected or echoed back from objects, and these are thought to be received by an oil filled channel in the lower jaw and conducted to the middle ear of the animal.

The Sperm Whale (*Physeter macrocephalus*) is a magnificent species of toothed whale. The species has been observed at all latitudes (see, for example, [1]), including in Australian waters. Individuals are known to dive to depths exceeding 3000 m in search of food, often squid. You can discover more about these awesome animals using the Species Profile and Threats (SPRAT) database [5] maintained by the Department of Environment and Energy.

4.2 Echolocation

When an animal uses echolocation, it emits a sound and waits for any reflected sound. An object is detected by the sound it reflects back to the animal. The time between the emission of a sound and the detection of an echo is called the *return time*. One theory is that the animal uses the return time of a sound to judge the distance to the object reflecting the sound.

In order for an animal to use the return time of a sound to judge the distance to an object reflecting the sound, the animal must "know" the speed of sound in the medium occupying the space between themselves and the object. The speed of sound is determined by the physical properties of the medium through which the sound wave is propagating. It is therefore reasonable to assume that sound takes the same length of time to travel from the animal to the object, as it does to travel back from the object to the animal. It follows that the distance to the object reflecting the sound is the distance travelled by sound in half the return time.

For example, suppose that an animal using echolocation in the ocean detects an echo 0.620 seconds after emitting a sound. If sound travels at a constant speed of $1503\ \mathrm{m} \cdot \mathrm{s}^{-1}$ in the seawater encompassing both the animal emitting the sound and the object reflecting the sound, then the distance between the animal and the object is

$$\begin{aligned} &\text{distance to the object detected using echolocation} \\ &= \frac{\text{return time}}{2} \times \text{speed of sound} \\ &= \frac{(0.620\ \mathrm{s})}{2} \times (1503\ \mathrm{m} \cdot \mathrm{s}^{-1}) \\ &\approx 466\ \mathrm{m}. \end{aligned}$$

4.3 The speed of sound in seawater

The speed of sound in seawater varies throughout the ocean. It has been found to depend mainly on three factors: the temperature of the water, the salinity of the water, and the depth at which the sound wave is propagating. There are several models for predicting the speed of sound in seawater given data on temperature, salinity and depth. One of the most simple is Mackenzie's Equation [3], first published in 1981. This model predicts that the speed of sound in seawater is given by

$$c = 1448.96 + 4.591T - 5.304 \times 10^{-2}T^2 + 2.374 \times 10^{-4}T^3 + 1.340(S - 35) + 1.630 \times 10^{-2}D + 1.675 \times 10^{-7}D^2 - 1.025 \times 10^{-2}T(S - 35) - 7.139 \times 10^{-13}TD^3,$$

where: c is the speed of sound in seawater in metres per second; T is the temperature of the seawater in degrees Celcius; S is the salinity of the seawater in parts per thousand (this is the amount of salt, in grams, per 1000 grams of seawater); and D is the depth, in metres, at which the sound wave is propagating. Mackenzie claims that the model is valid provided that the temperature is between -2 °C and 30 °C, the salinity is between 30 parts per thousand and 40 parts per thousand, and the depth is between 0 m and 8000 m.

4.4 Latitudes

A latitude is an angle that specifies how far away from the equator a point on the surface of the Earth is. Points on the equator have a latitude of 0°, the poles have latitudes of 90°. There are two different conventions used to distinguish between points with latitude 30° in the Northern Hemisphere and points with latitude 30° in the Southern Hemisphere. Some people specify the hemisphere with a letter after the latitude, writing N for North or S for South; others specify the hemisphere with a sign before the latitude, writing $+$ for North and $-$ for South. For example, the latitude of Brisbane is written $27.4698°$ S, or $-27.4698°$. Museum staff prefer the $\pm$ convention. Please note that when following this convention, the latitude of a point is a value from the interval $[-90, 90]$. It is good practice to explicitly write the sign of the latitude, except for points on the equator; that is, the latitude $27.4698°$ N may be written $+27.4698°$, and the latitude of a point on the equator may be written $0°$.

4.5 The salinity of seawater

The salinity of seawater varies, with typical values between 34 parts per thousand and 36 parts per thousand [2]. The salinity is generally lower at the poles than the equator. We shall assume that salinity is a quadratic function of latitude, being 36 parts per thousand at the equator and 34 parts per thousand at the poles.

4.6 The temperature of seawater at different depths and latitudes

Just as the atmosphere may be divided into layers characterized by how the temperature changes as altitude increases, the oceans may be divided into zones characterized by how the temperature changes as depth increases. We shall divide the oceans into three zones: the *surface zone* comprises the water at depths between 0 m and 200 m; the *thermocline* comprises the water at depths between 200 m and 1000 m; and the *deep zone* comprises water at depths exceeding 1000 m.

We shall assume that temperature is unaffected by longitude.

We shall assume that at all latitudes, temperature is a continuous function of depth. Informally, this means that you can sketch the graph of temperature versus depth (at a particular latitude and longitude) without lifting your pencil from the page.

We shall assume that the wind and waves serve to mix water in the surface zone so effectively that the temperature remains constant with depth. It does change with latitude, though, as the temperature in the surface zone is greatly affected by solar radiation [4]. We shall assume that in late March (around the time of the equinox) the temperature in the surface zone is a linear function of the absolute value of the latitude. Further, we shall assume that in late March the temperature in the surface zone is 2 °C at the poles, and 24 °C at the equator.

We shall also assume that, because solar energy never makes it to the deepest water, the temperature remains constant with depth in the deep zone. In fact, we shall assume that, no matter the latitude, the temperature of water in the deep zone in late March is 2 °C.

Since water in the surface zone can be warm, water in the deep zone is always cold, and the temperature at a given latitude is a continuous function of depth, the temperature must change with depth throughout the thermocline. We shall assume that, at each latitude and longitude, temperature is a linear function of depth in this zone.

5 Specifications for deliverable (D1)

Deliverable (D1) is an interactive Python program which models certain aspects of echolocation.

1. The museum staff have supplied a flow chart describing how the program should run. It is included on Page 7 of this document. Your code must be an implementation of the flow chart provided.
2. All messages to the user, including prompts to enter data, should communicate in a manner appropriate for the level of patron and should serve the purpose of the program. You may write different messages for patrons at different levels when you think it is appropriate to do so. You may indicate reasonable ranges for data entry when prompting the user. Writing appropriate messages is a way that you can demonstrate your communication skills.
3. You should use units appropriately in your communication with the user.
4. Any graphs that you display should be appropriately labeled, and may be accompanied by explanatory text. Providing a well-labelled graph is a way that you can demonstrate your communication skills.
5. Patrons will be able to enter data using a number keypad only, so all input will be numerical (the user can access only digits, the negative sign, a decimal point and the ENTER key).
6. You should use comments in your code to help the museum staff who may need to maintain and modify the code. Writing good documentation in your code is a way that you can demonstrate your communication skills.
7. Museum staff have a beginner's level of experience using Python, which you may regard as the equivalent of a student who has taken SCIE1100. If you write code that goes beyond that covered in the SCIE1100, you must provide comments that explain the code sufficiently well that museum staff can maintain it.
8. Museum staff are familiar with the library pylab, but they are not familiar with and have not installed other libraries. If you decide to include other libraries, you should include comments in the code to justify this choice and provide instructions for how staff at the museum can download the libraries.
9. You should provide a bibliography. This should be printed to the screen at the end of the program as part of the farewell message. Any standard referencing style is acceptable. The aim is to provide enough information to effectively acknowledge your sources. For example, you may refer to this task sheet as:

   SCIE1100 Python and Communication Task Sheet (2020). Faculty of Science, University of Queensland.
10. Museum staff have identified several functions that they think will be useful in possible modifications and extensions of the code. You must define these functions in your code, with the exact names specified below and so that they take the same arguments in the order specified. You should call these functions in your code as appropriate. You may define other new functions as you think appropriate.
    (a) You must define a function called

    `mackenzie_sos`

    which takes three arguments, representing the temperature (in degrees Celsius), depth (in metres) and salinity (in parts per thousand) in that order, and returns the speed of sound in seawater (in metres per second) according to Mackenzie's model.

    (b) You must define a function called

    `sal_of_seawater`

    which takes one argument, latitude (as a value in the interval $[-90, 90]$), and returns the salinity of seawater (in parts per thousand) at that latitude according to the model.

(c) You must define a function called

```
temp_of_seawater
```

which takes two arguments, latitude (as a value in the interval $[-90, 90]$) and depth (in metres) in that order, and returns the temperature of seawater in late March (in degrees Celsius) under those conditions according to the model.

11. Your code should be saved as a .py file. The file should be called

```
InteractiveEcholocationProgram********.py
```

with the string $********$ replaced by your student number.

6 Specifications for deliverable (D2)

Deliverable (D2) is a .pdf file containing sample output which demonstrates the functionality and features of your program. The file must contain sample output from running your code at least once for each patron type, but may contain more than that if necessary. The purposes of this document include: so that museum staff believe that you have delivered an item which meets the specifications; so that museum staff know what to expect when they run your program. The output from your test should be saved as a .pdf file. The file should be called

```
InteractiveEcholocationOutput********.pdf
```

with the string $********$ replaced by your student number.

References

[1] P.N. Halpin, A.J. Read, E. Fujioka, B.D. Best, B. Donnelly, L.J. Hazen, C. Kot, K. Urian, E. LaBrecque, A. Dimatteo, J. Cleary, C. Good, L.B. Crowder, and K.D. Hyrenbach. OBIS-SEAMAP. http://seamap.env.duke.edu/species/180488, 2009. Retrieved 21 August, 2018.

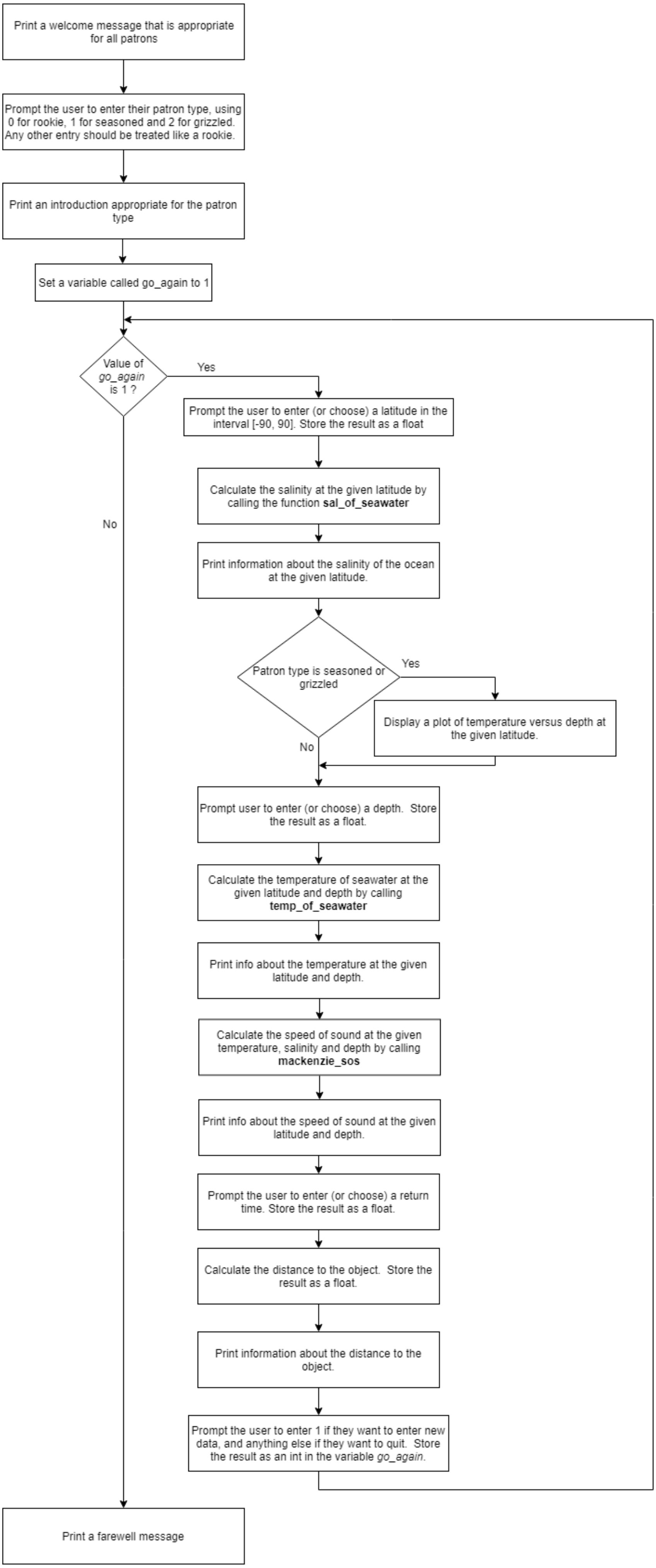
[2] Science Learning Hub. Ocean salinity. https://www.sciencelearn.org.nz/resources/686-ocean-salinity, 2018. Retrieved 21 August, 2018.

[3] Kenneth V. MacKenzie. Nine-term equation for sound speed in the ocean. *J. Acoust. Soc. Amer.*, 70:807–812, 09 1981.

[4] NOAA. How far does light travel in the ocean? National Ocean Service website. https://oceanservice.noaa.gov/facts/light_travel.html, 2018. Retrieved 24 August, 2018.

[5] Department of Energy and Environment. Species Profile and Threats Database. http://www.environment.gov.au/cgi-bin/sprat/public/sprat.pl, 2018.

[6] Department of Energy and Environment. Whales, dolphins and sound. http://www.environment.gov.au/marine/marine-species/cetaceans/whale-dolphins-sound, 2018. Retrieved 20 August, 2018.


Print a welcome message that is appropriate for all patrons
Prompt the user to enter their patron type, using 0 for rookie, 1 for seasoned and 2 for grizzled. Any other entry should be treated like a rookie.
Print an introduction appropriate for the patron type
Set a variable called go_again to 1
Value of go_again is 1 ?
Yes
No
Prompt the user to enter (or choose) a latitude in the interval [-90, 90]. Store the result as a float
Calculate the salinity at the given latitude by calling the function sal_of_seawater
Print information about the salinity of the ocean at the given latitude.
Patron type is seasoned or grizzled
Yes
Display a plot of temperature versus depth at the given latitude.
No
Prompt user to enter (or choose) a depth. Store the result as a float.
Calculate the temperature of seawater at the given latitude and depth by calling temp_of_seawater
Print info about the temperature at the given latitude and depth.
Calculate the speed of sound at the given temperature, salinity and depth by calling mackenzie_sos
Print info about the speed of sound at the given latitude and depth.
Prompt the user to enter (or choose) a return time. Store the result as a float.
Calculate the distance to the object. Store the result as a float.
Print information about the distance to the object.
Prompt the user to enter 1 if they want to enter new data, and anything else if they want to quit. Store the result as an int in the variable go_again.
Print a farewell message

Your assignment will be assigned the grade corresponding to the column that best describes your assignment. In the event that no single column best describes your assignment, a judgement will be made on the balance of the criteria, with criteria becoming more important as you move down the page.

Question	**mark = 7**	**mark = 6**	**mark = 5**	**mark = 4**	**mark = 3**	**mark (2, 1 or 0)**
How well is the code commented?	There is sufficient commenting. It is clear and accurate.	There is sufficient commenting. It is clear and accurate.	There is sufficient commenting. It is clear and accurate.	There is sufficient commenting.	There is commenting in the code.	Does not meet the criteria for a 3
Are the computations executed correctly?	Conceptual understanding of the mathematical models has been demonstrated and accurately applied.	Conceptual understanding of the mathematical models has been demonstrated and accurately applied.	Conceptual understanding of the mathematical models has been demonstrated and accurately applied.	Conceptual understanding of the mathematical models has mostly been demonstrated.	Limited conceptual understanding of the mathematical models has been demonstrated.	
Several principles of good programming practice concerning the use of variables and functions are described in the course. Are they adhered to in the code?	Variables names are informative, new functions and their arguments are always named, defined and called appropriately.	Variables names are informative, new functions and their arguments are always named, defined and called appropriately.	But for some minor violations, the principles of good programming discussed in the course are adhered to.	But for some minor violations, the principles of good programming discussed in the course are adhered to.	The principles of good programming practice are violated several times, or in an egregious way.	
Does the code run and meet the specifications?	The code runs without error, and the output file demonstrates that it meets the specifications, for all target audiences and expected inputs.	The code runs without error, and the output file demonstrates that it meets the specifications, for all target audiences and expected inputs.	The code runs without error and meets the specifications for all target audiences and expected inputs.	The code runs without error for most expected inputs. It meets specifications in all but perhaps a few minor ways.	The code runs for most expected inputs.	
What level of mastery of the principles of scientific communication is demonstrated in the output of the program?	The communication of scientific information and ideas is clear, fluent, and uses a level and style appropriate for each target audience. It displays insight and originality.	The communication of scientific information and ideas is clear, fluent, and uses a level and style appropriate for each target audience.	The communication of scientific information and ideas is mostly clear, fluent, and uses a level and style appropriate for each target audience.	The communication of scientific information and ideas is adequate for most target audiences.	The communication of scientific information and ideas lacks clarity or is not at the appropriate level for most target audiences. The communication may include information that demonstrates a scientific misunderstanding.	
How likely is this program to achieve its (highly ambitious) purpose? (A holistic assessment)	The program is very likely to achieve its purpose for all patrons.	The program is likely to achieve its purpose for most patrons.	The program may achieve its purpose for some patrons.			

This rubric has been constructed with reference to, and in compliance with, the UQ grade descriptors specified at http://ppl.app.uq.edu.au/content/3.10.07-grading-system