SCIE1000 Summer Semester, 2021
Python and Communication Assignment

1 The scenario

A public science museum in St Lucia is planning to update its exhibit. A feature of the museum is that exhibit items are 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.

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 early high school; 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.

Table 1: Characteristics of different patrons.

The museum is planning an exhibition called "Mother Nature, racing to survive." The topic is animal biomechanics, and the aim is summarised in the following passage from the exhibition prospectus:

> *With this exhibition we aim to instil in our patrons a sense of wonder at the amazing power and stamina displayed by animals as they chase their prey or avoid capture. Patrons will learn about the speeds at which animals can accelerate and run, and will be able to compare such motion with the fastest humans on Earth.*

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 beginner's 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 Mathematical Methods (intermediate high school mathematics); writing Python programs, including those which use arrays, loops, graphs and new functions; and communicating scientific information to various audiences.

2 An overview of the task

You will write an interactive text-based Python program that will run on a computer in the exhibition hall at the new science museum, as part of this exhibition. Your program will guide users to a better understanding and appreciation of how animals on the Serengeti plains can accelerate from rest to capture their prey or avoid being captured. The information you need to create the relevant models is provided in Section 5 of this document, and a high-level overview of how to complete the task is provided in Section 6.

This assignment requires you to produce and deliver two items:

(D1) A Python code file that satisfies the specifications in Section 7. This includes following the logical flow laid out in the flow chart provided in Figure 1 (see Page 10).

(D2) An audio-video screen capture file (3-4 minutes long) in which you show your code and communicate the overview of your approach to museum staff who will need to maintain your code.

3 Submission and grading

Both deliverables (D1) and (D2) are to be uploaded by **2pm on 28 January 2022** via a Blackboard submission link. Late submissions without an approved extension will be penalised according to the advertised policy; consult Section 5.3 of the Electronic Course Profile for more information.

Your submitted code will be run and tested as part of this grading process. A rubric (grading criteria) for this assignment is on Page 11. The file that you submit will be checked using software which is specially designed to detect plagiarism in code. Consult Section 6.1 of the ECP for more information and procedures concerning plagiarism.

Your assignment will separately be graded on two aspects, each on a 1–7 scale. The first aspect will be the way that you use the Python commands taught in SCIE1000 to implement 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 the communication within your program (for staff at the exhibit), the communication you use with the patrons of the exhibit, and the communication of your approach in audio-video screen capture.

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 more information is provided in Section 5.6. If you have any questions, please contact the course teaching staff via the course discussion board (see Section 4 below).

4 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 completing this assignment, but that help must be limited to general advice about modelling, Python and communication. 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.**

Remember that you must not look at anyone else's code and you must not show your code to anyone. Both actions are examples of behaviour that may be considered academic misconduct. No code from your assignment attempt should be posted on the course discussion board, or any other site, at any time. However, if you have problems with your code, you may develop some generic sample code that demonstrates the issue that you are having (but does not relate to the assignment). This can be discussed with others and/or posted to the discussion board for assistance. All such discussion board posts should be made visible to all students, so that everyone can see the question and the answer from the teaching staff.

Warning: In almost every preceding semester, the teaching team has initiated academic misconduct allegations against a small number of students who have completed this assignment (often leading to a grade of zero for the assignment). This has come about because students have failed to heed the above advice. The plagiarism detection software that we use is sophisticated and able to identify where students have worked together. Please complete all the work alone (don't sit with anybody while completing your assignment). Do not share your code (this will be treated the same as copying somebody else's code). This will minimise the stress and workload both for you and for the teaching staff. If you have questions about this aspect of the assignment, then contact the teaching team before starting.

5 Modelling the motion of mammals

5.1 Animals on the Serengeti

According to the United Nations Educational, Scientific and Cultural Organisation (UNESCO)[1]:

> *The vast plains of the Serengeti comprise 1.5 million ha of savannah. The annual migration to permanent water holes of vast herds of herbivores (wildebeest, gazelles and zebras), followed by their predators, is one of the most impressive natural events in the world.*

The aim of the museum exhibition is to explain how both predator and prey can accelerate to chase/avoid their adversary. The museum director would like you to write a Python program that allows patrons to see how quickly these animals can accelerate and run. Your program will calculate the motion of such animals and show a comparison with the fastest humans over a distance of 100 m. Your program will also model a chase between a predator and a prey (advanced section).

5.2 Maximum running speed of mammals

The maximal running speed of a mammal on the Serengeti could mean the difference between life and death. This maximal running speed can be considered to be a function of both the mass and the strength of the animal [2]. The masses of adult big cats on the Serengeti vary from around 40 kg for a leopard up to 140 kg for a hunting lioness. Both of these predators can maintain a maximum speed of 60 km/h. With a maximum speed of 110 km/h and an average mass of 75 kg, the cheetah is the fastest land mammal on the Serengeti. There are other predators on the Serengeti such as the hyena which can reach speeds of 50 km/h with an adult mass of about 45 kg. Herbivores on the Serengeti can be both much smaller and much larger than their carnivorous counterparts with masses ranging from 20 kg for the Thomson's gazelle, to 400 kg for a zebra and 500 kg for the massive African buffalo. As you might expect, the top speed of the gazelle is higher than that of the buffalo, with speeds of 65 km/h and 50 km/h respectively. Zebras can reach a top speed of 65 km/h. The most likely animal to be able to escape the clutches of a predator is the wildebeest with a weight of 250 kg and a surprising top speed of 80 km/h.

Although it is possible to develop a model of maximum running speed as a function of mass (see [2]), the accuracy of the model is insufficient to faithfully reproduce the speeds for the animals discussed above. Your program will thus need to provide the user with some guidelines for possible animal masses and maximum running speeds, and ask the user to input each of these values.

5.3 Acceleration of mammals

Neither predators nor prey can reach their top speed instantaneously, nor can they keep up their top speed indefinitely. In [3], it was found that the time taken for an animal to reach its top speed from rest, $t_{\max}$, is dependent on its mass, M, and that the model that best fits this relation is a power law of the form

$$t_{\max} = aM^b.$$

Here a and b are numerical constants. To find their values, the data provided in Table 2 should be used.

$\log_{10}(M/1\text{ kg})$	$\log_{10}(t_{\max}/1\text{ s})$
1.48	0.16
1.78	0.41
2.08	0.67
2.38	0.93
2.68	1.18

Table 2: Acceleration data for mammals.

5.4 Modelling speed and position

We will assume that the animal starts from rest, accelerates to top speed and then continues running at the top speed for a specified time. We will also assume that all motion is in a straight line. While accelerating, we will assume that the speed of the animal increases from zero to the maximum running speed, $v_{\max}$, in a linear fashion. Hence at time, t, we can describe the speed, v, of the animal as

$$v = \begin{cases} \frac{t \cdot v_{\max}}{t_{\max}} & t < t_{\max} \\ v_{\max} & t \geq t_{\max} \end{cases}$$

The calculation for the position will be done every 0.1 s for a specified time period. At each time step, we can calculate the next position using the equation

$$\text{New position} = \text{Current position} + \text{Current speed} \times 0.1.$$

Here appropriate units should be used for position and speed, noting that time is measured in seconds. Values for the time and position should be stored in arrays to allow graphing of the outcome.

5.5 Fastest humans

The museum director would like you to give context to your calculations by making a comparison between the motion of a mammal on the Serengeti with the fastest human sprinters. The world record holders for the 100 m are Usain Bolt (fastest male, 9.58 s, in 2009) and Florence Griffith-Joyner (fastest female, 10.49 s, in 1988). In an analysis of sprints at the 2009 World Championships in Athletics, the split times for each 10 m of a 100 m sprint by Usain Bolt were measured [4]. These are shown in Table 3.

Position (m)	0	10	20	30	40	50	60	70	80	90	100
Time (s)	0	1.88	2.88	3.78	4.64	5.47	6.29	7.10	7.92	8.87	9.58

Table 3: Split times for Usain Bolt in a 100 m race in 2009 [4].

The museum director asked a previous student (who had not taken SCIE1000) to develop a Python program to model the race using some of the theory introduced above (see page 9). The student identified that Usain Bolt's split times could be well modelled by assuming that he accelerated to a top speed of 43.3 km/h in 2.5 s. They have provided you with the student's code for your perusal in developing your own code. They would like you to compare the motion of animals on the Serengeti to the measured values for Usain Bolt given in Table 3. This means that the calculations that you do should be for a duration of 10 s, and comparisons should be made between the position of the mammal and the position of a sprinter for a duration of 10 s.

Important: the supplied code mostly **does not meet** the expectations for Python programming in this course. While you should view it and use it to understand the modelling, it should not form the basis for your own code (although you can certainly adapt some of the approaches for your own purposes).

5.6 Advanced section - modelling a pursuit

On the Serengeti, the relationship between a predator and its prey is well established. Each has well-developed skills optimised for survival. The museum staff would like you to model one such pursuit and identify whether the predator catches its prey, based on an initial separation. You can assume that predators can run at their top speed for only a limited time period. After this, they slow down at the same rate at which they accelerated and then stop. Prey can be assumed to be able to run at their top speed indefinitely. You should inform the patron about the outcome of the chase in a manner appropriate for each patron type.

6 A detailed overview of the task

Your assignment submission must follow the specifications listed in Section 7. Below, we give a high-level overview of how to approach the main section and the advanced section.

To complete the main section, you will need to:

- Find an appropriate mathematical function to model the relationship between animal mass and the time that this animal takes to reach its maximum speed. This must be done using the approaches introduced in SCIE1000 (i.e. you must not use software packages or techniques that have not been used in SCIE1000).
- You will need to write your own Python functions that implement particular aspects of the modelling (such as, but not limited to, the time to reach maximum speed).
- According to the main section of the flow chart:
  - Print information and ask for user input related to the contexts of animal mass and top running speed in a manner accessible for the patron type.
  - Use your model to calculate the motion of the mammal for a period of 10 s.
  - Print information about the outcomes of the model for the given input, and make comparisons of position with the fastest human after running for 10 s.
  - Provide a graph (with a description) for the enthusiast showing the position of the animal as a function of time. The graph should include the data for the fastest human for comparison.
- Include a description of how you approached this section of your code in your screen capture video (D2), including (briefly) how you developed your models and the overall code structure.

To complete the advanced section, you will need to:

- Choose one particular predator and one particular prey only - this is your choice. You could choose from those listed earlier in this document or find others of your choice (see [2] for more examples). Your program only needs to consider this predator/prey combination - you do not (and should not) ask the patron to choose these.
- For your chosen pair, identify their typical mass and maximum running speed.
- Ask the patron to input an initial separation between predator and prey.

- Calculate the motion of the predator and prey over a particular time frame. You will need to decide what that time frame is. It will depend on how long you consider the predator can maintain its top speed.
- Inform the patron of the outcome (in a manner appropriate for each patron type).
- Include a description of the approach you have used in your screen capture video.

7 Specifications for your submitted files

Specifications about the Python:

- Museum staff have supplied a **flowchart** describing how the program should run (Figure 1 on Page 10). Your code must be an implementation of the flowchart provided.
- You must 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. If you have any doubts about the use of a particular command, then put a (public) post on the discussion board.
- Your code should accurately reproduce the modelling that has been described in this document.
- Your code must be well-structured and follow the guidelines for programming practice, as introduced in SCIE1000. It should demonstrate your ability to use the approaches that have been introduced in the course including variables, input, conditionals, loops, arrays, functions and graphs.

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 several sentences to a (short paragraph) 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!
- Whenever you prompt the user for information, you may assume that they enter an appropriate number, and you can store their answer as a **float**. You do not need to check for incorrect inputs. Only numerical inputs are allowed (other types have not been covered in this course).
- You should **use units appropriately** in your communication with the user. Make sure the units of the values being passed into functions and the units of values being returned from functions are clear.

- 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**.

- Your **screen capture video should provide a clear overview of how your code works (including the modelling) and why you made the choices you did**. This does not replace excellent commenting in the code. One way to create such a file is by opening a Zoom meeting, sharing your screen, and selecting Record → Record to this computer.

- If you choose to use sources other than those cited in this task sheet, you should include a bibliography as comments at the end of your code, to show the museum staff maintaining the code where you obtained any relevant information you used. Any referencing style is fine.

File type and file name:

- Your assignment (D1) should be saved as a **.py file** called **Serengeti********.py** with the string ******** replaced by your student number.

- Your screen capture audio/video file (D2) should be saved as **Explanation********.mp4** with the string ******** replaced by your student number.

Important: In preparing and submitting your files

- First clear the kernel or exit and then re-enter your programming environment (Jupyter/Spyder/other) and then run your program. This ensures that all variables previously used will be cleared before your program is run. Correct any errors that appear. This replicates how your program will be tested.

- If you are using Jupyter, ensure that you save your file as a .py file (File - Download as - Python .py). After saving, open the file using a text editor (such as Microsoft Word) and check that all of your code appears in the exported file. Do NOT make any changes or save the code again from the text editor. If you need to make changes then go back to the version that you have in Jupyter.

- Check that your screen capture file plays properly and that the audio is clear.

- After you upload your files to Blackboard, check that both the Python file and the screen capture file have been uploaded. Ensure that you submit the files (not Save Draft). If your screen capture file is large, then your browser may appear to not be responding after you press the submit button. Wait for the files to upload (this could even take minutes if your internet is slow).

Sample code

Sample code to calculate the motion of a 100 m sprinter and compare with the split time of Usain Bolt during the 2009 World Championships in Athletics [4]. Note that this code mostly does not follow the principles of coding that have been introduced in SCIE1000.

```
# Code to calculate 100 m race
# x-distance, t-time, v-speed

from pylab import *
# Data
tB = array([0, 1.88, 2.88, 3.78, 4.64, 5.47, 6.29, 7.10, 7.92, 8.87, 9.58])
xB = array([0, 10, 20, 30, 40, 50, 60, 70, 80, 90, 100])
# Model
tM = 2.5
vM = 12
# Arrays - time step size 0.1
dt = 0.1
t = arange(0, 10.1, 0.1)
x = zeros(int(101))
# Calculate
i = 1
while i <= 100:
    if t[i] < tM:
        v = vM * t[i] / tM
    else:
        v = vM
    x[i] = x[i-1] + v * dt
    i = i + 1
# Plot
plot(x, t)
plot(xB,tB,"x")
show()
```


References

[1] UNESCO World Heritage Centre. Serengeti National Park. https://whc.unesco.org/en/list/156/. Accessed on 3/12/2021.

[2] Garland Jr. T. (1983) The relation between maximal running speed and body mass in terrestrial mammals. *Journal of Zoology*, 199:157–170.

[3] Myriam R. Hirt, Walter Jetz, Björn C. Rall, Ulrich Brose (2017) A general scaling law reveals why the largest animals are not the fastest. *Nature Ecology and Evolution*, 1(8):1116–1122.

[4] Grauber R., Nixdorf E. (2011) Biomechanical Analysis of the Sprint and Hurdles Events at the 2009 IAAF World Championships in Athletics. *New Studies in Athletics* 26:1/2, 19-53.

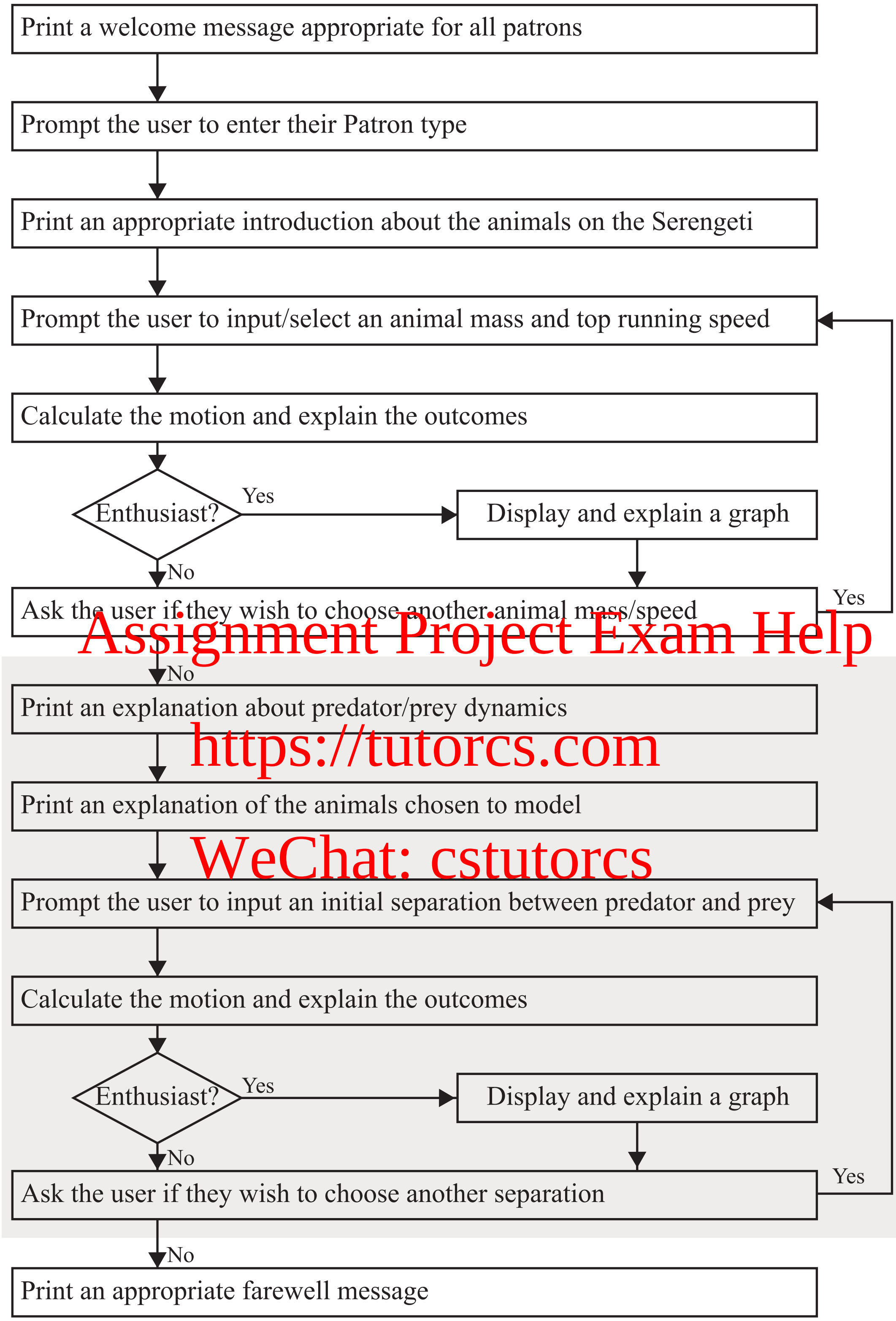

Figure 1: Flowchart for the interactive program (shaded section indicates the advanced section).

Python and Communication Assignment Grading Criteria

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. 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 only 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 (or will run after the correction of a minor error), 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, the screen capture video, and to patrons) mostly meets the specifications. There 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 (within the code, the screen capture video, and to patrons) is mostly clear, fluent, and largely meets all the specifications.
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 specifications are mostly met.	The level of communication is adequate throughout the advanced section but would require some editing before use in an exhibit. The advanced section is also communicated in the screen capture video and mostly meets the specifications.
+2	The code for the advanced section runs without error and accurately represents the modelling. It meets all the required specifications and shows insight in the development of the model.	The level of communication is clear throughout the advanced section and meets all the specifications. The advanced section is also clearly communicated in the screen capture video.