# Solent University

# Coursework Assessment Brief

# Assessment Details

|  |  |
| --- | --- |
| Unit Title: | Problem Solving through Programming |
| Unit Code: | COM404 |
| Unit Leader: | Prins Butt |
| Level: | 4 |
| Assessment Title: | Python Fundamentals |
| Assessment Number: | 2 |
| Assessment Type: | TCA |
| Restrictions on Time/Word Count: | 2 hours |
| Consequence of not meeting time/word count limit: | It is essential that assignments keep within the time/word count limit stated above. Any work beyond the maximum time/word length permitted will be disregarded and not accounted for in the final grade. |
| Individual/Group: | Individual |
| Assessment Weighting: | 50% |
| Issue Date: | 10th December 2018 |
| Hand In Date: | 14th December 2018 |
| Planned Feedback Date: | 1st February 2018 (4 working weeks after hand-in) |
| Mode of Submission: | On-line (via ‘Solent Online Learning’) |
| Number of copies to be submitted: | 1 |
| Anonymous Marking | This assessment will be marked anonymously |

# Assessment Task

# The purpose of this time-constrained assessment is to assess your understanding of key event-driven programming concepts and their application in problem solving.

You are expected to attempt the questions in this time-constrained assessment within a two hour invigilated session.

You are expected to code your solutions using the Python 3 programming language and submit your solutions online via Solent Online Learning using the links provided and as directed on the unit page.

**Part A: Graphical User Interface (~40 minutes)**

Stan wishes to create a graphical user interface for a newsletter subscription. The intended graphical user interface is shown in figure below:

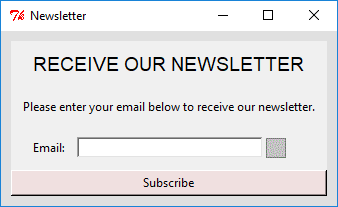


**Task A1 (30 marks):** Implement the above design using Python 3 and Tkinter. It is suggested that you save your resulting python file as part\_a.py. Please check your solution against the following marking criteria before submitting:

**Marks for this question are allocated as follows:**

* The properties of the **window** are set as follows:
  + 1 mark: The window's title is set to "Newsletter"
  + 1 mark: The window's background colour is set to "#eee"
  + 1 mark: The window has a padding of 10 pixels all the way round
* The properties of the **heading label** are set as follows:
  + 1 mark: The label's text is set to "RECEIVE OUR NEWSLETTER"
  + 1 mark: The label fills the available horizontal space
  + 1 mark: The label's font is set to "Arial" of size 14
  + 1 mark: The label has a padding of 10 pixels at the top and bottom
* The properties of the **instructional label** are set as follows:
  + 1 mark: The label's text is set to "Please enter your email below to receive our newsletter."
  + 1 mark: The label fills the available horizontal space
  + 1 mark: The label's text is left justified
  + 1 mark: The label has a padding of 10 pixels all the way round
* The properties of the **email label** are set as follows:
  + 1 mark: The label's text is set to "Email:"
  + 1 mark: The label has a padding of 10 pixels on its left and right side
* The properties of the **email entry** are set as follows:
  + 1 mark: The entry's border has a thickness of 2 pixels
  + 1 mark: The entry's font colour is "#f00"
  + 1 mark: The entry has a width of 30 pixels
  + 1 mark: The entry is displayed next to the email label
* The properties of the **subscribe button** are set as follows:
  + 1 mark: The button's text is set to "Subscribe"
  + 1 mark: The button fills the available horizontal space
  + 1 mark: The button's background colour is set to "#fee"
* Appropriate use of layout components and layout managers:
  + 5 marks: The appropriate use of frame(s) for grouping and styling
  + 5 marks: Layout matches the design in the figure above

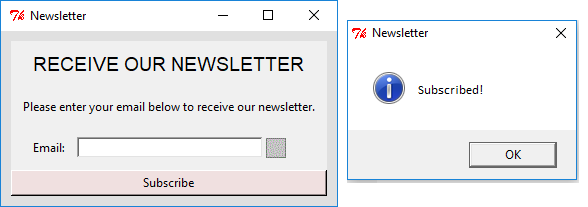
**Task A2 (8 marks):** Stan wishes to add an image next to the email entry which for now shows the [default.gif](https://learn.solent.ac.uk/pluginfile.php/1271526/mod_label/intro/default.gif) image *(right click the link to save the image)* as shown in the figure below:

  
It is suggested that you add to the code in your existing part\_a.py file when addressing this task. Please check your solution against the following marking criteria before submitting:

**Marks for this question are allocated as follows:**

* 2 marks: The [default.gif](https://learn.solent.ac.uk/pluginfile.php/1271526/mod_label/intro/default.gif) image has been loaded
* 3 marks: The loaded image is displayed in an image label
* 1 mark: The image label has a padding of 10 horizontal pixels
* 2 marks: The image label is placed to the right of the email entry

**Task A3 (7 marks):** When the subscribe button is clicked a message box should be displayed which displays a "Subscribed!" message as indicated in the figure below:



It is suggested that you add to the code in your existing part\_a.py file when addressing this task. Please check your solution against the following marking criteria before submitting:

**Marks for this question are allocated as follows:**

* 1 marks: The subscribe button is bound to the ButtonRelease-1event
* 3 marks: A suitable user defined function has been added to handle the event
* 1 marks: A message box is displayed when the subscribe button is clicked
* 1 mark: The message box has the title "Newsletter"
* 1 mark: The message box shows the message "Subscribed!"

Once your solution is ready please upload it using the link below. Note: you are only uploading your part\_a.py file at this stage. You should not click the submit button until you have attempted all parts of this TCA:

[Upload your solution for this question here](https://learn.solent.ac.uk/mod/assign/view.php?id=797620)

**Part B: Functionality (~40 minutes)**

Note: You should save your part\_a.py from the previous question as part\_b.py and use this when attempting Part B.

Stan wishes to add some functionality to his newsletter program so that it displays [filled.gif](https://learn.solent.ac.uk/pluginfile.php/1272442/mod_label/intro/filled.gif) or [empty.gif](https://learn.solent.ac.uk/pluginfile.php/1272442/mod_label/intro/empty.gif) *(right click each link to download the images)*when the contents of the email entry are modified as shown in the figure below:



**Task B1 (10 marks):** Implement the above design using Python 3 and Tkinter. It is suggested that you save your resulting python file as part\_b.py. Please check your solution against the following marking criteria before submitting:

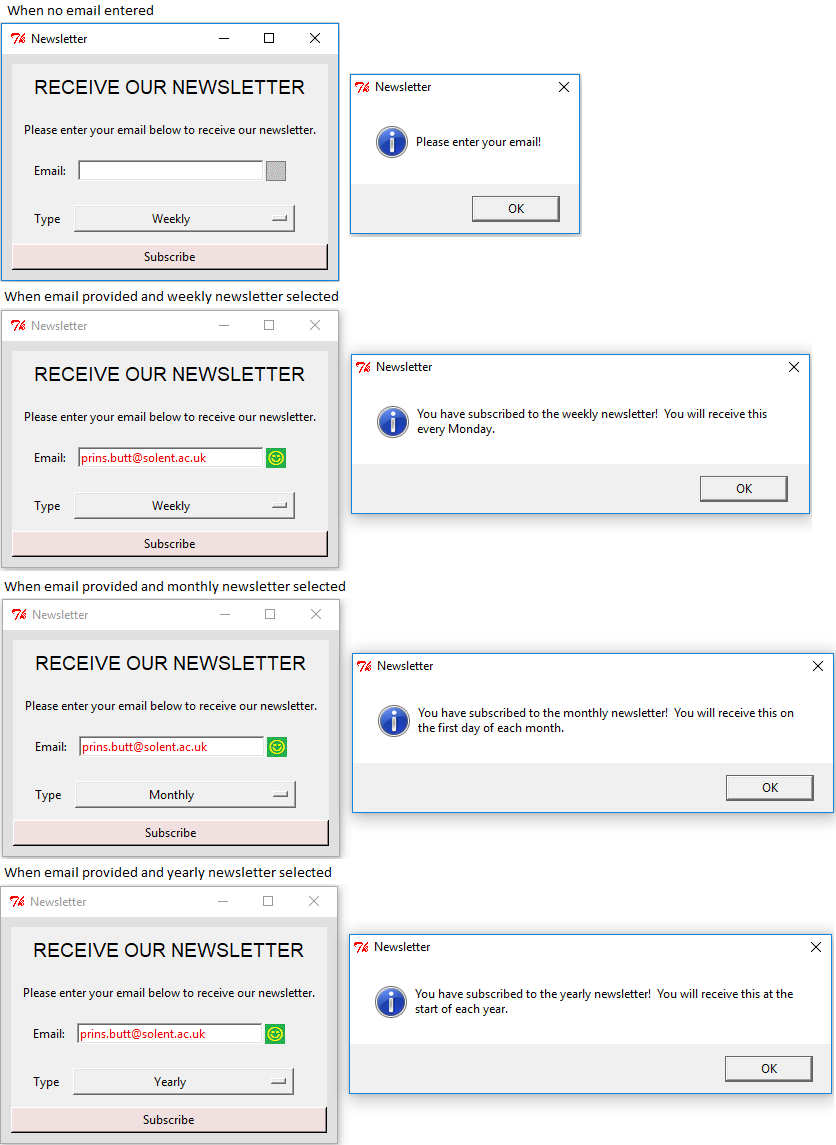
**Marks for this question are allocated as follows:**

* 2 marks: The provided [filled.gif](https://learn.solent.ac.uk/pluginfile.php/1272442/mod_label/intro/filled.gif) and empty.gif have been loaded
* 1 mark: The email entry has been bound to the "<KeyRelease>"event
* 1 mark: The [default.gif](https://learn.solent.ac.uk/pluginfile.php/1272442/mod_label/intro/default.gif) image is displayed when the window first loads
* 3 marks: The filled.gif image is displayed when text is entered in the email entry
* 3 marks: The [empty.gif](https://learn.solent.ac.uk/pluginfile.php/1272442/mod_label/intro/empty.gif) image is displayed when all text is removed from the email entry

**Task B2 (10 marks):** Stan wishes to extend his newsletter program so that users can select how often (Weekly, Monthly or Yearly) to receive the newsletter as indicated in the figure below:



When the subscribe button is clicked one of four messages should be displayed depending on whether an email is provided and the type of newsletter that has been selected as illustrated by figure below:

  
It is suggested that you add to the code in your existing part\_b.py file when addressing this task. Please check your solution against the following marking criteria before submitting:

**Marks for this question are allocated as follows:**

* 1 marks: Display an email required message when no email is provided and subscribed is clicked
* 3 marks: Display weekly newsletter message when an email is provided and the weekly newsletter is selected
* 3 marks: Display monthly newsletter message when an email is provided and the monthly newsletter is selected
* 3 marks: Display yearly newsletter message when an email is provided and the yearly newsletter is selected

Once your solution is ready please upload it using the link below. Note: you are only uploading your part\_b.py file at this stage. You should not click the submit button until you have attempted all parts of this TCA:

[Upload your solution for this question here](https://learn.solent.ac.uk/mod/assign/view.php?id=797620)

**Part C: Animation (~40 minutes)**

Note: You should save your part\_b.py from the previous question as part\_c.py and use this when attempting Part C.

Stan wishes to add some animation to his newsletter program. In the first instance he would like to add an animate button which displays the text "Start Animation". With each click of the button the text of the button should alternate between "Stop Animation" and "Start Animation" as illustrated in the figure below:

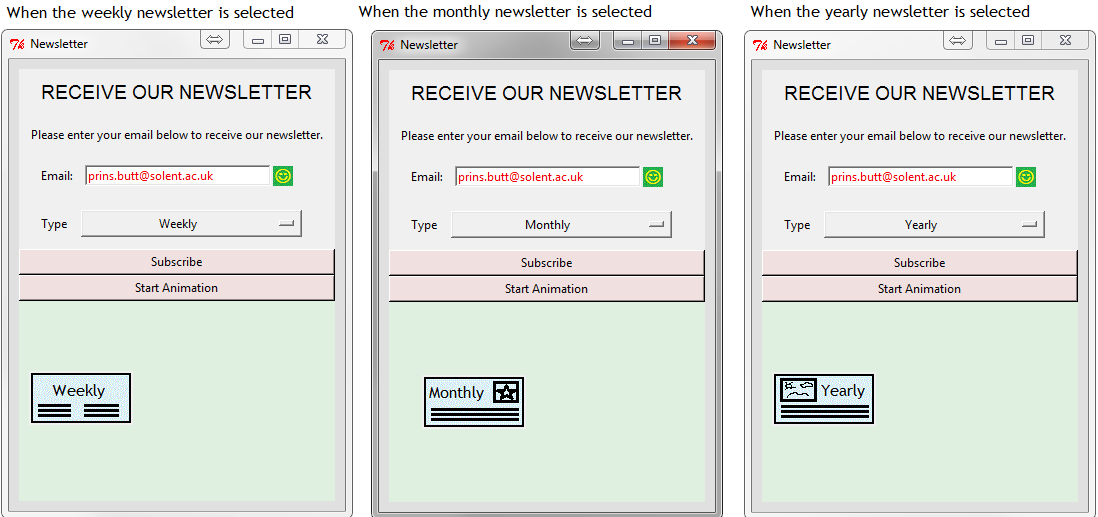


**Task C1 (4 marks):**Implement the above animate button using Python 3 and Tkinter. Please check your solution against the following marking criteria before submitting:

**Marks for this question are allocated as follows:**

* 1 mark: Added an animate button
* 1 mark: Added a variable to track the state of the animation (running or not running)
* 2 marks: Changed the text of the animate button to the appropriate value when clicked

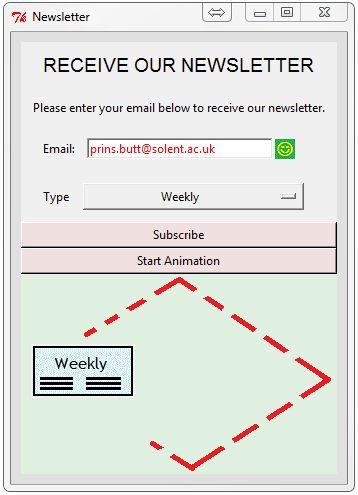
**Task C2 (9 marks):**Stan wishes to display a picture of a newsletter in his animation. The picture should change each time the animate button is clicked. The picture displayed (one of [weekly.gif](https://learn.solent.ac.uk/pluginfile.php/1272582/mod_label/intro/weekly.gif), [monthly.gif](https://learn.solent.ac.uk/pluginfile.php/1272582/mod_label/intro/monthly.gif) or [yearly.gif](https://learn.solent.ac.uk/pluginfile.php/1272582/mod_label/intro/yearly.gif)) will depend on the type of newsletter selected as indicated in the figure below:



**Marks for this question are allocated as follows:**

* 2 marks: Added a frame that fills the available horizontal space and is 200 pixels in height
* 1 mark: Placed an image in the frame
* 2 mark: The weekly.gif image is displayed when the animate button is clicked and the weekly newsletter is selected
* 2 marks: The monthly.gif image is displayed when the animate button is clicked and the monthly newsletter is selected
* 2 marks: The yearly.gif image is displayed when the animate button is clicked and the yearly newsletter is selected

**Task C3 (22 marks):**Stan wishes to animate his newsletter image so that it bounces of the sides of the animation frame as indicated by the red line in the figure below:



**Marks for this question are allocated as follows:**

* 2 marks: Added variables to track the position of the image
* 2 marks: Added variables to control the direction of movement of the image
* 4 marks: The image bounces of the top of animation frame
* 4 marks: The image bounces of the bottom of the animation frame
* 4 marks: The image bounces of the left side of the animation frame
* 4 marks: The image bounces of the right side of the animation frame
* 2 marks: The image stays within the bounds of the animation frame when bouncing around the frame.

Once your solution is ready please upload it using the link below. You should click the submit button when you are ready to submit your final solution:

[Upload your solution for this question here](https://learn.solent.ac.uk/mod/assign/view.php?id=797620)

# Assessment criteria

|  |  |
| --- | --- |
| **Grade** | **Criteria** |
| **A** | In addition to the criteria below, demonstrates problem solving and understanding of constructing a graphical user interface that go beyond taught concepts. |
| **B** | In addition to the criteria below, demonstrates understanding of constructing a graphical user interface that can utilise a range of resources, constructs and handle both system and user driven events. |
| **C** | In addition to the criteria below, demonstrates understanding of constructing a graphical user interface that can handle a range of user initiated events and contains nested logic structures. |
| **D** | Demonstrates understanding of constructing a graphical user interface consisting of simple event handling. |
| **F** | Does not reach the expected threshold for a pass grade. |

# Learning Outcomes

This assessment will enable students to demonstrate in full or in part the learning outcomes identified in the unit descriptors.

# Late Submissions

Students are reminded that:

1. If this assessment is submitted late i.e. within 5 working days of the submission deadline, the mark will be capped at 40% if a pass mark is achieved;
2. If this assessment is submitted later than 5 working days after the submission deadline, the work will be regarded as a non-submission and will be awarded a zero;
3. If this assessment is being submitted as a referred piece of work then it must be submitted by the deadline date; any Refer assessment submitted late will be regarded as a non-submission and will be awarded a zero.

<http://portal.solent.ac.uk/documents/academic-services/academic-handbook/section-2/2o-assessment-principles-and-regulations.pdf?t=1534423842941>

# Extenuating Circumstances

The University’s Extenuating Circumstances procedure is in place if there are genuine circumstances that may prevent a student submitting an assessment. If students are not 'fit to study’, they can either request an extension to the submission deadline of 5 working days or they can request to submit the assessment at the next opportunity (Defer). In both instances students must submit an EC application with relevant evidence. If accepted by the EC Panel there will be no academic penalty for late submission or non-submission dependent on what is requested. Students are reminded that EC covers only short term issues (20 working days) and that if they experience longer term matters that impact on learning then they must contact the Student Hub for advice.

A summary of guidance notes for students is given below:

<http://portal.solent.ac.uk/documents/academic-services/academic-handbook/section-2/2p-extenuating-circumstances.pdf?t=1534423896787>

# Academic Misconduct

Any submission must be students’ own work and, where facts or ideas have been used from other sources, these sources must be appropriately referenced. The University’s Academic Handbook includes the definitions of all practices that will be deemed to constitute academic misconduct. Students should check this link before submitting their work.

Procedures relating to student academic misconduct are given below:

<http://portal.solent.ac.uk/support/official-documents/information-for-students/complaints-conduct/student-academic-misconduct.aspx>

**Ethics Policy**

The work being carried out by students must be in compliance with the Ethics Policy. Where there is an ethical issue, as specified within the Ethics Policy, then students will need an ethics release or an ethical approval prior to the start of the project.

The Ethics Policy is contained within Section 2S of the Academic Handbook:

<http://portal.solent.ac.uk/documents/academic-services/academic-handbook/section-2/2s-university-ethics-policy.pdf>

**Grade marking**

The University uses a letter grade scale for the marking of assessments. Unless students have been specifically informed otherwise their marked assignment will be awarded a letter grade. More detailed information on grade marking and the grade scale can be found on the portal and in the Student Handbook.

<http://portal.solent.ac.uk/documents/academic-services/academic-handbook/section-2/2o-annex-2-assessment-regulations-grade-marking-scale.pdf?t=1534424273208>

**Guidance for online submission through Solent Online Learning (SOL)**

<http://learn.solent.ac.uk/onlinesubmission>