|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Task:** | | **2** | | |
| **Task Title:** | | **Portfolio** | | |
| **Task Code:** | | **AT2 POR-Task-2** | | |
|  | |  | | |
| Assessment type (): | | | | |
|  | Questioning (Oral/Written) | |  | Portfolio |
|  | Practical Demonstration | |  | Project |
|  | 3rd Party Report | |  | Other – Please Specify |

|  |
| --- |
| The base requirements this assessment task include:   * Web server, Python interpreter and database server * IDE or editor for developing Python programs (only PyCharm supported by the college) * Raspberry Pi with SenseHat * Access to Office 365 & Microsoft Word * Report Template (Portfolio: Part 1 Document Template) as supplied   Use of some of these items may not occur in this part of the assessment task. |
| Assessment Due This assessment is split into components that have several due dates:   * Week 07 17:00 (5:00PM) on the day of the scheduled lecture. * Week 09 17:00 (5:00PM) on the day of the scheduled lecture. * Week 12 17:00 (5:00PM) on the day of the scheduled lecture. * Week 14 17:00 (5:00PM) on the day of the scheduled lecture.   Refer to Blackboard for most accurate dates, which may alter due to unforeseen circumstances.  We also will endeavour to update these document(s) at the same time. |
| Instructions Follow the steps listed in this assessment item. Please note that **additional** information may be given.  Submission of the documentation, code, and associated items is at the end of each part of the portfolio.  Each part of the portfolio has a deadline for submission.  It is advantageous to you to attempt to meet the deadline provided. |
| Important If you are using a different configuration of tools and equipment for this assessment item, then assistance in this and subsequent parts of the portfolio to ensure the systems work correctly will be limited. |
| Scenario You are currently working as a junior software developer at CUBE Music Pty Ltd, a Perth-based company that creates specialised music player software.  The company is looking into adding features that will allow smart multi-room music playback, including some form of “follow me”, which will use various IoT sensors.  You have decided to do some more analysis of the requirements while at the same time coding a few prototyping solutions of the smaller problems you identify. This will help you to get a better feel for the subject matter and will guide you to a solution. |
| General Instructions We provide a document template for your answers.  Save the file as:   * XXX-IoT-Port-Part-2.docx   Replacing the XXX with your initials.  For example, Adrian Gould would use AG-IoT-Port-Part-2.docx for his submitted filename.  Upload any code as a PyCharm project in a zip-file. Remove the virtual environment (**venv** or **.venv**) from the zip-file before uploading it to Blackboard. |
| Answering Questions When a step includes a question, you must attempt to answer it.  There is a minimum and maximum number of words to use for each answer.  If a step has more than one question, these maxima and minima are a total for all the questions in that specific step.  All answers must be in complete sentences unless indicated. |
| Sources of Information In industry, it is good practice to keep track of where information was obtained. This is especially true if it is a written document, or even code.  If you answer any questions using information from web sites, please include the site name and URL (Web site address) after the answer. Likewise, include the title and author for books and magazine articles. For example:   * RS Electronics Ltd: <https://au.rs-online.com/> * Slack API Documentation, Users List Method: <https://api.slack.com/methods/users.list>  Code Storage We advise that you create a GIT repository on GitHub and use this to store a copy of your work.  You may also use OneDrive within your college Office365 to store a backup of your code or keep a copy on a USB thumb drive. |
| *This space left intentionally blank.* |

| **STEP** | **Task to perform** | Words Min/Max |
| --- | --- | --- |
| 00 | Create Evidence Document Make sure you have followed the instructions on creating the answer document, as given in the General Instructions.  Familiarise yourself with the content and document your progress in this assessment.  Make sure that you complete the title page of the document.  At any stage during this assignment, you may consult the stakeholder(s) or their representative(s). |  |
|  | *This space left intentionally blank.* |  |
| 01 | Create a new PyCharm project Create a new project in PyCharm with the name **XXX-IOT-POR-Task-2**, where **XXX** should be replaced with your initials. Create a new virtual environment in the project and ensure that you are using Python3. You do not have to create a “welcome” script **main.py** yet. | n/a |
|  | *This space left intentionally blank.* |  |
| 02 | Create an ‘event’ class From the requirements, you infer that a core element of the solution might be an ‘event’. This will indicate that someone is (or isn’t) in a room or a zone.  Since your prototype will use a Raspberry Pi with SenseHat, you should incorporate at least one extra sensor value in that class. (This is also in the requirements.)  In your project, create a file called **event.py**.  In this file, create a class called **Event** with the following instance variables:   * location (string) * timestamp (string, or datetime object) * presence (Boolean) * sensor value(s) *\*)*   *\*) Pick at least one sensor value from the SenseHat and give the instance variable a sensible name. If you add more than one, provide an instance variable for each one.*  Don’t forget to add a docstring to the class, so you know what this class is about in two- or three-weeks’ time (and as a courtesy to other people who may need to read your code).  Make sure to follow the PEP-8 conventions with regards to naming instances variables and other coding style issues. | n/a |
|  | *This space left intentionally blank.* |  |
| 03 | Printing human readable info Add the method **\_\_str\_\_** to your new Event class. This should return a string that contains a human-readable version of the information contained in the class (or rather, the object).  Think about what you would like the output to look like when you print an object of type **Event** and implement the aforementioned method.  Explain your choice. | n/a |
|  | The print out needs to display not only the returned values, but also tell the user what these values are ie return f”location: (self.location),…” will display location: room1.  It would also be good to display the output of each item returned on a separate line  *This space left intentionally blank.* |  |
| 04 | Simple testing and inspection Add the following code to the file **event.py** (at the bottom):  **if \_\_name\_\_ == ‘\_\_main\_\_’:**  ***# Create some events and inspect their outputs***  **pass**  Ensure the indentation is correct.  Replace the pass with some code that:   * instantiates at least two different Event objects. * prints those objects so you can inspect the output. * changes the value of **presence** for one event. * prints that object again and inspect it shows the new value.   Run the file **event.py** as a script.  Grab a screenshot of the output and put that in the answer box below. | n/a |
|  | *This space left intentionally blank.* |  |
| 05 | Events are, well, events You have realised that you are only interested in changes. If a person enters a room (**presence == True**), they stay there until they leave (**presence == False**).  You’ve decided to collect events but to only store events that have a different value for **presence** from the last one for a single room.  At this point, there is no need for persistent storage (i.e., database).  In your project, create a file called **location\_events.py**. In this file, create a class called **LocationEvents**.  When you instantiate this object, you must pass in a location, which will be stored as an instance variable. You must also create an empty list of potential events as an instance variable. Write the **\_\_init\_\_** method so that these requirements are met. |  |
|  | *This space left intentionally blank.* |  |
| 06 | Analysis: Class Diagram While the requirements to add events to the new class aren’t incredibly complicated, you think it’s worthwhile to analyse this a bit further. You have decided that now is a good time to create a few UML diagrams.  So far, you have created two classes, **Event** and **LocationEvents**.  Draw a Class Diagram that shows the relationship between these classes. Use the correct type of association between them and show multiplicities. |  |
|  | *This space left intentionally blank.* |  |
| 07 | Analysis: Adding events Looking at the Class Diagram and thinking about the requirements, you quickly realise that the class **LocationEvents** should have some method to add Events to it.  Update the Class Diagram to reflect this. The method should take an **Event** as an argument and return a Boolean to indicate failure/success. Make sure the Class Diagram correctly reflects the method’s prototype. |  |
|  | *This space left intentionally blank.* |  |
| 08 | Analysis: Sequence Diagram (success) With the knowledge you now have about the classes, you feel this is a good time to write a sequence diagram to check your assumptions.  The Sequence Diagram should show: a “user” (which could be another system), the **LocationEvents** class and the **Event** class.  Your Sequence Diagram should show the following (success) scenario:   1. The “user” creates a **LocationEvents** object for a certain location 2. The user then creates an **Event** for that location 3. The user adds the event to the **LocationEvents** object by calling the relevant method (graphically) 4. The method returns **True** to indicate success   Put the Sequence Diagram in the answer box below. You do not have to write any additional code at this point. |  |
|  | Flow: Event: Person enter\_room = True  Call Event: Get\_location, set old room presence false, set new room presence true, get sensors |  |
| 09 | Sequence Diagram (alternate scenarios) Looking at the Sequence Diagram, you realise that the implementation will be more complicated because of the following requirements:   * An event should only be stored if the value for **presence** is different from the last value stored, and * An event should only be stored if the location matches the location of the **LocationEvents** container object.   Update your Sequence Diagram to show these two scenarios by using **Event**s that have the same value as the previous one or have the wrong location.  Alternatively, you can create a new Sequence Diagram for each scenario if you feel that’s easier or clearer.  Put the updated Sequence Diagram(s) in the answer box below. |  |
|  | *This space left intentionally blank.* |  |
| 10 | Implement the method Based on your analysis, you now think you have enough information to implement the method that you identified earlier.  Add a new method to the class **LocationEvents** and give it a meaningful name. Use the method prototype that you identified earlier.  Write a few lines of docstring to explain what the method does and what the only argument means.  Implement the code that:   * adds a new **Event** to the list (if it is valid), or * refuses to add it (if it is not valid).   Make sure the method returns **True** or **False**, respectively. |  |
|  |  |  |
| 11 | Keep the project in a safe location Please keep your project in a safe location, as you’ll need the code for the next part(s) of the Portfolio Assessment. |  |
|  | Submission of Portfolio Work To submit the portfolio, do the following:   * Save the document with your answers as a MS Word file (.docx). * Open Blackboard, and locate the AT2 Portfolio Task 1 assessment * Open the assessment and upload the original word-processed document. * Click submit.   Whilst there is no need to use any other word processing software as you have access to Office 365 for free as a student, if you use Apple Pages, or Open Office, we will then require you to upload the original file **AND** a PDF version. |  |

# Appendix A: Code Style Guidelines

Your code will follow the PEP 8 standard.

Readability Counts  
- Zen of Python

Explicit is better than implicit.  
- Zen of Python

Other code standards available in the Presentation, “Python Coding Standards for North Metropolitan TAFE”.