**ASSIGNMENT**

**Procedural Programming Term 1**

Ro-Ro-Ro-Your-Bots™

For this assignment the first step of my original plan was to read the brief thoroughly and make notes.

Later on, I realised the consequences of my having neglected to carry this out in depth (see below).

Because I was procrastinating actually diving into actually beginning the assignment proper, my next step was to have a shot at learning the stipulated referencing system, APA 7th, on the university library website.

In actuality, I knew that I didn’t know how to start the coding.

My thinking was that I needed to decide how the program would interact with the user:

Visually, how would I display the status of the robots and workers when all I knew how to print out using Python was a single line at a time?

My original idea was that of attempting to learn how to use a Python graphics package. Online, I saw that popular ones appeared to be curses and tkinter, although the pandas data-handling package would also probably have been sufficient for this project.

I found myself feeling daunted at the prospect of learning how to use one of these tools in addition to sharpening my basic Python skills enough to handling the back-end side.

After discussion of it with a couple of the student mentors from the School of Computing and Engineering, and, even more influential, with my module tutor, it was suggested that I focus on the fundamentals.

While I attempted to get my head round how I was going to handle the various programming challenges, I amused myself with naming the project and sketching a humorous storyline for it, influenced by the cartoon "Futurama" with its slightly dark irony.

(Cohen, Groening, Keeler, Katz, 1999 – present)

I was unsure about how the user would control what was happening in the scenario, but since I wasn't going to use a GUI, it was going to have to be by textual commands.

I wanted to provide as much guidance for the user as I could, rather than making a program that would mimic the functioning of one in an actual factory, where the supervisor would build up a familiarity with the software, and would need minimal onscreen instruction.

- Main functions

* *learn Python* ***STYLE*** *to format code :*
* *Use PEP8 style guide for Python -* [*https://peps.python.org/pep-0008/*](https://peps.python.org/pep-0008/)
* *learn how to format essays at University of Huddersfield:*
  + *line spacing*
  + *font*
  + *alignment*
  + *page numbering*
* *formatting code*

WHAT I DID:

Watched YouTube Python tutorial on classes - <https://www.youtube.com/watch?v=ZDa-Z5JzLYM&pp=ygUOcHl0aG9uIGNsYXNzZXM%3D>

Established I was going to use classes to store the information for each robot and human.

Created functions to add or remove robots and humans. Because the functions were as yet empty, I included the pass instruction in them so that the program would skip over them without raising errors.

For each function or class I created I included comments, both to remind me of what the code was for, and also to divide it up so I could read it more easily.

I began to work with a method of keeping a back-up file storing the previous version of the file. Only when I was satisfied that the changes I had made since then were going to work correctly did I update the main version of the file. Then I learnt the basics of Git logging.

I used the scratch files function in PyCharm to test isolated sections of code I was working on without having to run the entire program.

I started reading up on articles about how to allow the program to function in real time and on how to use classes.

Then I found out that the brief had been altered so that real time functionality was no longer required and that no classes were to be used, so I amended what I had written to reflect these changes.

The first code I wrote was the intro, then the function for the user to input how many robots and workers they want. I had to sanitise the input as best as I could, then print a response.

*Difficulties encountered:*

initial "How many robots do you want?"

- input control / sanitization :

- printing out workforce loop

list or dictionary ?

- read the brief more carefully

setting up dictionaries  
 - especially doing both robots and humans in one function

- then getting only the keys printed in a loop

working out how to use dictionaries:

how to define them using a loop

condensing functions together, making helper functions

control flow – calling functions from other functions

- idea to use "while True" infinite loops

- seemed to work better using while not loops testing for the presence of a variable initialised just to serve as a conditional

much ado about global and local functions

- security v. readability – reuse variable names in different settings?

I found it very difficult to remove characters from a string, where I was trying to get the user to enter a list separated by commas

5/1/2025

Freaked out massively when I realised I needed the dictionaries of workers and robots to indicate their IDLE / WORKING / FINISHED status, not the tasks each were working on. I had it to include 12 status codes to include each task, but it threw out the function for the user to change their statuses manually, as that would discard the data about which task they were on (which needs updating manually separately from their status.

I had to go through the whole program changing task codes for status codes.

6/1/2025

Got confused trying to make a time calculator that could format a gross amount of seconds into HH/MM/SS . The calculations kept going way off, arriving at the wrong number of minutes in a day, too far off to be attributed to Python's dodgy decimal divisions. Looking again in the morning I realised it was all down to a typo; I had inputted a 5 instead of a 6.

7/1/2025

Realised the print layout I’d been so proud of won’t display on the Command Prompt terminals installed on the university computers; they don’t accept the same format of escape codes as on my own laptop. For my "clear screen" function the impairment is only cosmetic, but when it comes to displaying the task log, printing it without the escape codes to position the cursor renders the data unreadable.

Also, the display style I'm using will only work on appropriately sized terminal windows. The advice online is to use the Curses package, so in terms of visual display I’m back where I started.

Thankfully my tutor isn’t reliant on the university computers to mark my project, otherwise I'd have to have either abandoned my clear visual layouts, printing it as bare text, or used massive and complex f-strings to recreate my tables a line at a time.

Another hitch: I updated PyCharm, which involved restarting it. When the new version booted up, it linked itself to my GitHub repository, but whereas I thought it would offer to let me update my repository from the version I'd been working with only minutes previously in the old IDE, it updated the version I was working on from the older draft I'd previously stored on GitHub. Thankfully I had already had a bit of practice with git reset, and was pleased indeed to refresh my memory of it.

Security:

Passing a big array like my task log between functions exposes too much data too frequently. I could have sectioned off individual variables or lists from it to pass to the functions as needed as arguments instead of the entire log.

Were this a real program enabling management of an industrial workplace, even just the worker IDs would be confidential. Companies have data protection standards they need to observe, and having the employees' personal information flying around the system would be reckless.

Tracking Task Progress:

I realised the table I'd concocted to track the progress of specific tasks wouldn't fit the criteria outlined in the brief. My table didn't make logical sense: how could it show the percentage each task was towards completion and the number of tasks in progress too, unless either it showed each instance of that task individually, or all the tasks had begun at the same time?

It meant me going back to the drawing board.

Conclusion

Reflecting honestly on my work during this project, I have emerged with several take-aways.

The first is, that in all truth, I need to read project briefs more thoroughly.

Perhaps if I had printed it out and annotated it, I would have had a clearer picture in my mind of how the program was going to have to function.

Some dead ends I went down could perhaps, in a parallel history, have been averted, such as forgetting I needed to store the simple form of the "WORKING " status of workers, in addition to which specific task they were working on.

The second is the rookie error of not backing up regularly. Especially once I had (thought) I had learnt how to use Git, I started to become reliant on this, not realising that without a thorough understanding of the complexity of Git, it would probably be easier to just make back-ups as .txt files, notwithstanding the kudos of having my program on GitHub.

The third is to try and make a flow diagram as part of the planning process. My program quickly became labyrinthine in its complexity, and it was hard to keep track mentally of how it would jump between functions, and the order they were all in.

I will have to swallow my pride and keep a pencil and pad of paper to hand, so that I can keep a record of the principal functions, variables and the alterations in the variable names as they are passed as arguments to the multiple parameter names of the many functions.

References

American Psychological Association. (2021, 2023, 2025). *Reference Examples*. <https://apastyle.apa.org/style-grammar-guidelines/references/examples>

Library Services. *APA 7th Referencing Guide.* University of Huddersfield. https://library.hud.ac.uk/pages/apareferencing/

Bell, A. (2018). *Get programming : learn to code with Python.* Manning Publications.

BowlOfRed. (2022). *Instead of escaping the newline, how about just using explicit quoting?* [Comment on the online forum post *Pythonic way of printing multi-line string*s]. Python. https://discuss.python.org/t/pythonic-way-of-printing-multi-line-strings/19681/3

Cohen, D.X.., Groening, M., Keeler, K., &Katz, C. (Executive Producers). (1999 – present). *Futurama* [TV series].  Fox Broadcasting Company; Comedy Central;  Hulu.

ConnorWill. (2022). *Ansi Escape Sequences cheatsheet.* GitHub. <https://gist.github.com/ConnerWill/d4b6c776b509add763e17f9f113fd25b>

Faccioni, J.L. (2022). *5 Ways to Create a Dictionary in Python*. <https://learnpython.com/blog/create-dictionary-in-python/>

Howell, A & Burns, S. (2023). *How to roll back Git code to a previous commit.* https://www.techtarget.com/searchitoperations/answer/How-to-roll-back-Git-code-to-a-previous-commit

Javatpoint. *Nested Tuples in Python*. https://www.javatpoint.com/nested-tuples-in-python

ModernMind. (2004). *Python Programming for Beginners Made Easy*. ModernMind Publications.

NKMK. (2023). *Get keys from a dictionary by value in Python*. https://note.nkmk.me/en/python-dict-get-key-from-value/

RealPython. (2017). *Python Function Argument Unpacking Tutorial (\* and \*\* Operators).* <https://www.youtube.com/watch?v=YWY4BZi_o28>

Ramos, LP. (2024). *Dictionaries in Python.* https://realpython.com/python-dicts/

Ronquillo, A (2019) . *A Beginner’s Guide to the Python time Module.* Retrieved November 9th, 2024 from <https://realpython.com/python-time-module/>

Schafer, C. (2016). *Python OOP Tutorial 1: Classes and Instances*. Retrieved November 9th, 2024 from <https://www.youtube.com/watch?v=ZDa-Z5JzLYM&pp=ygUOcHl0aG9uIGNsYXNzZXM%3D>

Schafer, C. (2017). *Python Tutorial for Beginners 5: Dictionaries - Working with Key-Value Pairs*. Retrieved Jan 31st, 2024 from <https://www.youtube.com/watch?v=daefaLgNkw0>

Stack Overflow community. (2018). *count the number of occurrences of a certain value in a dictionary in python?* Retrieved Jan 6th 2025.Found at https://stackoverflow.com/questions/48371856/count-the-number-of-occurrences-of-a-certain-value-in-a-dictionary-in-python

Stack Overflow community. (2009). *Git replacing LF with CRLF*. Retrieved Dec 29th, 2024 from <https://stackoverflow.com/questions/1967370/git-replacing-lf-with-crlf>

Stack Overflow community. (2008) [*How do I get my program to sleep for 50 milliseconds?*](https://stackoverflow.com/questions/377454/how-do-i-get-my-program-to-sleep-for-50-milliseconds)Retrieved Dec 30th, 2024. Found at <https://stackoverflow.com/questions/377454/how-do-i-get-my-program-to-sleep-for-50-milliseconds>

Stack Overflow community. (2014). *How do you get Python to detect for no input.* Retrieved Dec 31st, 2024. Found at <https://stackoverflow.com/questions/26226489/how-do-you-get-python-to-detect-for-no-input>

Stack Overflow community. (2020*) How do I print these hash symbols using a loop? [duplicate]*. Retrieved Jan 1st, 2025. Found at <https://stackoverflow.com/questions/64541968/how-do-i-print-these-hash-symbols-using-a-loop>

Stack Overflow community. (2011). *Linux Terminal Display and Python.* Retrieved Jan 7th, 2025. Found at

<https://stackoverflow.com/questions/7636984/linux-terminal-display-and-python>

Stack Overflow community. (2010, edited 2023). *Remove specific characters from a string in Python*. Retrieved Jan 3rd, 2025. <https://stackoverflow.com/questions/3939361/remove-specific-characters-from-a-string-in-python>

Woodring, J. (2023). *Using ANSI Escape Codes to Color Your Terminal Output.* Youtube video. Found at <https://www.youtube.com/watch?v=yQ9Ns6Z4Q-s>

W3Schools. <https://www.w3schools.com/python>