University of Roehampton London

LAST UPDATED: 16 NOVEMBER 2022 BY FAKHRELDIN SAEED, SHEKOUFEH RAHIMI AND ARTURO ARAUJO

SOFTWARE DEVELOPMENT 1 COURSEWORK SPECIFICATION

Coursework 2 will help you continue your journey into software development, building on the previous coursework 1. The focus here is to help you understand how you could write a program that repeats the same task multiple times, providing a scaffolding on how to encapsulate computational processes in functions. To complete this coursework, you will design the block of codes necessary to tackle the given problem. To evaluate progress and highlight any gaps in understanding, these problem sets will provide a set of challenging questions and programming situations. Time in lab, seminars and office hours may be used to clarify any concepts

Submission date: December 8 via CodeGrade in Moodle by 16:59 hours GMT.

Contribution: 30% of the total mark.

Aim: To gauge understanding on the correct use of variables and objects, data types and the ability to read/write data; as well as the develop students' fluency in programming languages and software development.

Learning Outcomes: 1: Design, implement, test, and debug a program that uses each of the following fundamental programming constructs – basic computation, simple I/O, standard conditional and iterative structures, the definition of functions, and parameter passing. 2: Analyse and explain the behaviour of simple programs involving the fundamental programming constructs – variables, expressions, assignments, I/O, control constructs, functions, parameter passing, and recursion. 3: Identify the relative strengths and weaknesses among multiple designs or implementations for a problem. 4: Use a programming language to implement, test, and debug algorithms for solving simple problems.

Grading Advice: While fully functioning programs and its documentation that meet all the requirements are the goal of the test, the programs and the rationale for their design, implementation, testing and debugging will be assessed for efficiency, simplicity, creativity, and good style (PEP-8). Please read the Grading Rubric and bring any questions you may have to your tutor.

COURSEWORK 2

To complete this coursework, you will design the behaviour of complex programs involving the fundamental programming constructs – functions, operations, and file reading/writing. A template Jupiter notebook and a text file are provided to complete the tasks. **Time in class can be used to troubleshoot and get feedback on your program.**

The prestigious football club who originally hired our Software Development company is very pleased with the outcome and has requested an extension. The program that you have crafted for coursework 1 takes in rates of 6 skills of new players in the form of a score between 0 and 5 (shouldn't take inputs outside of this range). The six main criteria to rate the skills of the player should be:

1. Speed
2. Shooting
3. Passing
4. Defending
5. Dribbling
6. Physicality

The program uses these individual skill rates to calculate an overall rating out of 100 using the formula:

*overall rate = summation of skills rate * 100 / 30*

The resulting number (should be between 0 and 100) is then used to estimate the salary using the salary dataset:

No	Salary	Overall rate
1	1000	80
2	700	60
3	500	45
4	400	30

From the dataset above,

- If the overall rate is greater than or equal **80**, the salary should be: **1000**
- If the overall rate less than or equal to 30, the salary should be: **400**
- If the overall rate is in between two numbers, like 45 and 30, the output should be a range of the maximum salary and the minimum salary: **500 400**

You are asked to add the following functionality (up to 78 points, depending on scoring an "Excellent" score in each rubric component):

1. **Functions (8 points):** Split and pack up your coursework 1 code into functions. The program must contain a main() function and at least two other functions called "calculate_rating()" and "calculate_salary()".
   Use the main() function to call the other functions to receive the 6 player's skill ratings, calculate the player's score and their salary range.

2. **Input expansion (8 points):** Adjust your program to ask users to enter a player ID (2-digit number), name and date-of-birth (D.o.B) by sequence, before asking for the 6 skills from coursework 1 (speed, shooting, passing, defending, dribbling, physicality). D.o.B should be in ISO format YEAR-MONTH-DATE.

3. **Loop your code (8 points):** Your program should use a loop to ask for the information of three players, or until the input for user ID is "end".

4. **Calculate the players' overall ratings and store it for later use (2 points).**

5. **Calculate the players' salary ranges and store it for later use (2 points).**

6. **Calculate players' ages (8 points):** According to each user's year of birth, your program should calculate the age of the player and store it for later use. Hint: the library datetime can be used here to make your job easier.

7. **Tabulate (8 points):** Display a table that summarises the input data (ascending order with user ID) using the tabulate library. It should look like:

```
  UID  Name                D.o.B         Age    Score  Salary Range
-----  ------------------  ----------  -----  -------  --------------
   13  Lieke Martens       1992-12-16     29  86.6667  1000
   02  Cristiano Ronaldo   1985-02-05     37  73.3333  1000 700
   24  Lucy Bronze         1991-10-28     31  50       700 500
```

8. **File writing (8 points):** Save the table into a new local file named "players.txt"
9. **Make sure that you:**
   a. **Implement robust input validation for a player's ID, D.o.B., and ratings.** Your program should give a warning that says ***"The rating you entered was invalid"*** for any input errors **(6 points)**
   b. **Display use of appropriate data structures for storing player information (5 points)**
   c. **Display use of appropriate code structures for implementing the functionality (5 points)**

10. **Distinction Advanced Function:** Once everything in requirements 1-9 has been completed, for a distinction grade, extend your program with one more function called advanced(). This function should:
    a. be a complimentary main() method that, instead of taking user input, uses the provided file as input.
    b. accept in its argument (filename) the name of the text file to read.
    c. read the user information record from the provided file "PlayerData.txt", which contains player IDs, names, and date-of-births. The function should then use this data as the input for the program.
    d. call all the other functions used and produces a text file output (requirement 8).

Submission Requirements: Submit your code via Codegrade as a Jupyter notebook named:

footballer_skills.ipynb

The program file should contain full documentation explaining your implementation (comment your code). Your file should have your name, ID number and date of last update.

Feedback: Feedback will be given automatically through Codegrade via Moodle.

Marking Scheme: Grades will be based on number of components completed. You will be assessed on

- **Solution:** the completeness of the code, documentation, and reflection to meet the specification given.
- **Design:** ability to decompose a problem into coherent and reusable parts.
- **Correctness:** ability to create solution that reliably produces correct answers or appropriate results.
- **Logic:** ability to use correct program structures appropriate to the problem domain.
- **Clarity:** ability to format and document code for human consumption (PEP-8) and reflection.
- **Robustness:** ability of the solution to handle unexpected input and error conditions correctly as evidenced via testing.

The following rubric will be used to assess your work:

Basic Requirements

In order to get full marks for this rubric category your code must:

- Be encapsulated into functions including `*main()*`, `*calculate_rating()*`, and `*calculate_salary()*`. (8)
- Prompt users for an ID, name, D.o.B, and 6 player ratings, in that order. (8)
- Accept data for exactly 3 players before stopping or if "end" was entered instead of an ID. (8)
- Calculate the players' overall ratings. (2)
- Calculate the players' salary ranges. (2)
- Calculate the players' ages. (8)
- Display a table of all the data passed to the program using `tabulate` (8)
- Save this table to `players.txt` (8)

Intermediate requirements

In order to get full marks for the rubric category your code must:

- implement robust input validation for a player's ID, D.o.B., and ratings (6)
- Display use of appropriate data structures for storing player information (5)
- Display use of appropriate code structures for implementing the functionality (5)

Distinction Requirements:

Distinction work in the range of 72-78 should demonstrate:

- exemplary attainment of learning outcomes
- a high level of insight and critical evaluation of the material
- a comprehensive and up-to-date account of relevant theoretical and empirical material
- a thorough understanding and integration of material supporting a cogent argument
- excellent writing of a high academic standard

Distinction work in the range of 82-100 should demonstrate:

- attainment beyond the intended learning outcomes
- an outstanding level of originality and creativity, providing a significant new perspective on the question or topic
- a clear, elegant and well supported argument, based on the integration and sophisticated critical evaluation of a substantial body of knowledge
- suitability for publication in high quality journal

Academic Misconduct: All submissions will be processed through a code plagiarism tool. If signs of misconduct are found, all students involved will be contacted to discuss further steps. Please see here for information on academic integrity at the university https://portal.roehampton.ac.uk/information/Pages/Academic-Integrity.aspx.

Our guiding principle is that academic integrity and honesty are fundamental to the academic work you produce at the University of Roehampton. You are expected to complete coursework which is your own and which is referenced appropriately. The university has in place measures to detect academic dishonesty in all its forms. If you are found to be cheating or attempting to gain an unfair advantage over other students in any way, this is considered academic misconduct and you will be penalised accordingly. Please don't do it.