COMP 1039

Problem Solving and Programming

Programming Assignment 2

UniSA STEM
University of South Australia
May, 2024

Contents

INTRODUCTION

This document describes the second assignment for Problem Solving and Programming.

The assignment is intended to provide you with the opportunity to put into practice what you have learnt in the course by applying your knowledge and skills to the implementation **a program that will maintain information on players playing blackjack (dice) against the computer.**

This assignment is an **individual task** that will require an **individual submission**. If you are an **internal student,** you will be required to submit your work via learnonline before **Tuesday 11 June (swot-vac), 10.00am.**

This document is a kind of specification of the required end product that will be generated by implementing the assignment. Like many specifications, it is written in English and hence will contain some imperfectly specified parts. Please make sure you seek clarification if you are not clear on any aspect of this assignment.

ASSIGNMENT OVERVIEW

Manage player information

You are required to write a Python program that will manage player information. The program will allow players to play blackjack (dice version) against the computer and will maintain information on players (using a List of Lists). The program will keep a record of the players who play blackjack (dice version). Player information will be stored in a text file that will be read in when the program commences. Once the initial player information has been read in from the file, the program should allow the user to interactively query and manipulate the player information as well as play blackjack against the computer. Please ensure that you read sections titled 'Assignment Specification' below for further details.

Please ensure that you read sections titled 'Assignment Specification' below for further details.

GRADUATE QUALITIES

By undertaking this assessment, you will progress in developing the qualities of a University of South Australia graduate.

The Graduate qualities being assessed by this assignment are:

- The ability to demonstrate and apply a body of knowledge (GQ1) gained from the lectures, workshops, practicals and readings. This is demonstrated in your ability to apply problem solving and programming theory to a practical situation.

- The development of skills required for lifelong learning (GQ2), by searching for information and learning to use and understand the resources provided (Python standard library, lectures, workshops, practical exercises, etc); in order to complete a programming exercise.

- The ability to effectively problem solve (GQ3) using Python to complete the programming problem. Effective problem solving is demonstrated by the ability to understand what is required, utilise the relevant information from lectures, workshops and practical work, write Python code, and evaluate the effectiveness of the code by testing it.

- The ability to work autonomously (GQ4) in order to complete the task.

- The use of communication skills (GQ6) by producing code that has been properly formatted; and writing adequate, concise and clear comments.

- The application of international standards (GQ7) by making sure your solution conforms to the standards presented in the Python Style Guide slides (available on the course website).

ASSIGNMENT SPECIFICATION – MANAGE PLAYER INFORMATION

Write a **menu driven program** called *assign2_yourEmailId*.py that will allow the user to enter commands and process these commands until the quit command is entered. The program will store and maintain player information (using a List of Lists). Player information will be stored in a text file that will be read in when the program commences. Once the initial player data has been read in from the file, the program should allow the user to interactively query and manipulate the player information.

Input

When your program begins, it will read in player information from a file called players.txt. This is a text file that stores information pertaining to players. An example input file called players.txt can be found on the course website (under the Assessment tab). You may assume that all data is in the correct format. The name of the player is stored on a separate line. The very next line contains the number of games played, games won, games lost, games drawn, chip balance and the total score. This information is stored on one line and is separated by the space character as seen in Figure 1 below:

After the program has stored the data (using a List of Lists), it will enter interactive mode as described in the following section.

```
Bruce Wayne
5 5 0 0 100 15
Jessica Jones
12 0 6 6 10 6
Johnny Rose
6 2 0 4 20 10
Gina Linetti
7 4 0 3 300 15
Buster Bluth
3 0 2 1 50 1
```

Figure 1: *Player information file format (players.txt).*

Your program will maintain **one List of Lists** as follows:

```
player_list = []    # List of Lists to store player information
```

Once the above information is read in from the file, the player list will be populated as follows:

	player_list
0	[Bruce Wayne, 5, 5, 0, 0, 100, 15]
1	[Jessica Jones, 12, 0, 6, 6, 10, 6]
2	[Johnny Rose, 6, 2, 0, 4, 20, 10]
3	[Gina Linetti, 7, 4, 0, 3, 300, 15]
4	[Buster Bluth, 3, 0, 2, 1, 50, 1]

Note: A player and their statistics are stored as a list and are stored as one item in the player list, i.e. a List of Lists (as seen above).

Interactive Mode

Your program should enter an interactive mode after the player information has been read from the file. The program will allow the user to enter commands and process these commands until the quit command is entered. The following commands should be allowed:

1. **list:**
   Displays for all players, the player's name, games played, won, lost, drawn, number of chips and their total score. Outputs the contents of the player list as seen below in the section titled Screen Format. Please read the section at the end of this handout titled – 'Useful Built-In Python Functions'.

2. **buy:**
   Prompts for and reads the player's name and searches for the player in the list of players (list of lists). If the player is found in the list of players, the number of chips the player would like to buy is prompted for and read (the player can only buy between 1-100 chips inclusive at a time). The player's chip balance is updated accordingly and a message indicating that this has been done is displayed to the screen. If the player is not found in the player list, an error message is displayed.

3. **search:**
   Prompts for and reads the player's name and searches for the player in the player list. If the player is found in the player list, the player's name, games played, won, lost, drawn, number of chips and their total score, are displayed to the screen as seen below (in the section titled Screen Format). If the player is not found in the list of players, an error message stating the player has not been found is displayed to the screen. Please read the section at the end of this handout titled – 'Useful Built-In Python Functions'.

4. **high:**
   Displays the player with the highest chip balance in the list of players. Where two players have the same chip balance, the player with the lower games played value should be displayed to the screen - see section titled Screen Format below. If no players are stored in the list or a player with a highest chip balance cannot be found (i.e. all players have a chip balance of zero), display an error message accordingly.

5. **add:**
   Prompts for and reads the player's name. If the player does not already exist (i.e. a match is not found on the player's name), the player is added to the list of players. If the player is added, chip balance is initialised to 100 and all other data members (games played, games won, games lost, games drawn, and the player's total score), are initialised to zero and a message is displayed to the screen indicating that this has been successfully added.

   The player information must be added after the last player entry stored in the list (i.e. at the end of the list). If the player is already stored in the player list, an error message is displayed. No duplicate entries are allowed.

6. **remove:**
   Prompts for the player's name. If the player is found, he/she is removed from the list of players and a message is displayed to the screen indicating that this has been done. If the player is not found in the player list, an error message is displayed.

7. **play:**
   Prompts for the player's name. The program searches for the player in the list of players and if the player is not found in the list of players, an error message is displayed to the screen.

   If the player is found in the list of players, he/she is then able to play Blackjack against the computer until he/she does not wish to continue playing (enters 'n' when prompted to 'Play again (y|n)?').

After every game, the player's game statistics are updated (i.e. games played, total score, etc…). Three (3) points are awarded if the player wins, zero (0) points are awarded if the player loses and one (1) point is awarded if the player draws with the dealer. The player's chip balance is also updated. If the player wins, the player's chip balance will be increased by the amount of chips bet. If the player loses, the player's chip balance will be decreased by the amount of chips bet. A push (draw) result does not change the player's chip balance in any way.

8. **chips:**
   Displays the list of players in descending order of chips. Where two players have the same number of chips, the player with the lower games played value should appear first. This command should not alter the original list of players in any way. The information is displayed to the screen as described in the section titled 'Screen Format' below.

9. **quit:**
   Causes the program to quit, outputting the contents of the player list (List of lists) to a file (see section '*Final Output*' below for format).

Note:

The program should display an appropriate message if a player is not found matching a search criteria. Appropriate messages should also be displayed to indicate whether a command has been successfully completed.

Please refer to the sample output (at the end of this handout) to ensure that your program is behaving correctly and that you have the correct output messages.

Each time your program prompts for a command, it should display the list of available commands. See the sample output (at the end of this handout) to ensure that you have the output format correct.

For example:

```
Please enter choice
[list, buy, search, high, add, remove, play, chips, quit]:
```

Menu input should be validated with an appropriate message being displayed if incorrect input is entered.

Screen Format

The **`list`** command (`display_players()` function) should display the player information in the following format:

```
============================================================
-                      Player Summary                      -
============================================================
-                              P  W  L  D   Chips   Score  -
------------------------------------------------------------
-  Bruce Wayne                 5  5  0  0     100      15  -
------------------------------------------------------------
-  Jessica Jones              12  0  6  6      10       6  -
------------------------------------------------------------
-  Johnny Rose                 6  2  0  4      20      10  -
------------------------------------------------------------
-  Gina Linetti                7  4  0  3     300      15  -
------------------------------------------------------------
-  Buster Bluth                3  0  2  1      50       1  -
------------------------------------------------------------
============================================================
```

Likewise for the **`chips`** command (which should use the same `display_players()` function) should display the player information (sorted on chips) in the following format:

```
============================================================
-                      Player Summary                      -
============================================================
-                              P  W  L  D   Chips   Score  -
------------------------------------------------------------
-  Gina Linetti                7  4  0  3     300      15  -
------------------------------------------------------------
-  Bruce Wayne                 5  5  0  0     100      15  -
------------------------------------------------------------
-  Buster Bluth                3  0  2  1      50       1  -
------------------------------------------------------------
-  Johnny Rose                 6  2  0  4      20      10  -
------------------------------------------------------------
-  Jessica Jones              12  0  6  6      10       6  -
------------------------------------------------------------
============================================================
```

The **`search`** command should display individual player information to the screen in the following format:

```
Johnny Rose stats:

P  W  L  D  Score
6  2  0  4  10

Chips:  20
```

The **`high`** command (`display_highest_chip_holder()` function) should display the player with the highest number of battles won in the following format:

```
Highest Chip Holder => Gina Linetti with 300 chips!
```

Final Output

When your program finishes (because you entered the quit command) your program should output the contents of the list of players to a file called `new_players.txt`.

The format of this file should **exactly** match that of the input file.

PRACTICAL REQUIREMENTS

It is recommended that you develop this part of the assignment in the suggested stages.

It is expected that your solution WILL include the use of:

- Your solution in a file called *assing2_yourEmailId*.py.
- The supplied `blackjack.py` module (that plays one game of Blackjack (dice) against the computer). This is provided for you – **do NOT modify this file**.
- Use of the supplied `play_one_game(no_chips)` function from the `blackjack.py` module. You are not required to implement the `play_one_game(no_chips)` function or the `blackjack.py` module. This is provided for you (blackjack.py module) – **do NOT modify this file**.
- Appropriate and well constructed `while` and/or `for` loops. (Marks will be lost if you use `break` statements or the like in order to exit from loops).
- A **list of lists** to store player information.
- You **must** implement **each** function listed below.
- Appropriate `if, if-else, if-elif-else` statements (as necessary).

- The following ten functions:
  - **`read_file(filename)`**

    This function takes a file name and reads the contents of that file into a list called player_list. The function returns the newly created list of players (i.e. a list of lists). You **must** use a loop in your solution. You **may** use String and/or List methods **in this function only**. You may find the String methods `split()` and `strip()` useful here. Please note: This function will be provided for you. You may use your own function or use the function provided… the decision is yours. :)
  - **`write_to_file(filename, player_list)`**

    This function will output the contents of the player list (list of list) to a file in the same format as the input file. The file will need to be opened for writing in this function (and of course closed once all writing has been done). The function accepts the filename of the file to write to and the list of players. You **must** use a loop in your solution.
  - **`display_players(player_list)`**

    This function will take the list of players (list of lists) as a parameter and will output the contents of the list to the screen. This function displays the information to the screen in the format specified in the assignment specifications under the section - 'Screen Format'. You **must** use a loop in your solution. Please have a read of the section at the end of this handout titled – 'Useful Built-In Python Functions'.
  - **`find_player(player_list, name)`**

    This function will take the player's name as input along with the list of players (player list of lists) and will return the position (index) of the player found in the player_list. If the player is not found, the function returns -1. You **must** use a loop in your solution. You **must not** use list methods in your solution.
  - **`buy_player_chips(player_list, name)`**

    This function takes the player list (list of lists) and the player's name and updates the player's chip balance. If the player is found, the function prompts for and reads the number of chips the player would like to buy (between 1-100) and updates the player's chip balance by that amount. A message indicating that this has been done is displayed to the screen. You should validate the number of chips to ensure the player only enters between 1-100. If the player is not found in the list of players, an error message is displayed to the screen. You **must** call function `find_player()` from this function.

- **`display_highest_chip_holder(player_list)`**

  This function takes the player list (list of lists) as a parameter and displays the player with the highest chip balance in the list of players to the screen. Where two players have the same chip balance, the player with the lower games played value should be displayed to the screen. If no players are stored in the list or a player with a highest chip balance cannot be found (i.e. all players have a chip balance of zero), display an error message accordingly.

- **`add_player(player_list, name)`**

  This function takes the list of players (list of lists) and the player's (to be added) name, as parameters. If the player already exists in the list of players, display an error message to the screen. If the player does not exist in the player list (list of lists), the player is added. Create a new list with the information read in (i.e. name) and add the player's information (as a list) to the end of the list of players (list of lists). If the player is added, chip balance is initialised to 100 and all other stats data (games played, no won, no lost, no drawn and the player's total score), are initialised to zero and a message is displayed to the screen indicating this has been successfully added. This function returns the list of players (list of lists). You may use the `list_name.append(item)` method to add the new player to the list of players. You **must** call function `find_player()` from this function.

- **`remove_player(player_list, name)`**

  This function takes the list of players (list of lists) and the player's (to be removed) name as parameters. If the player is not found in the list of players, display an error message to the screen. If the player does exist in the players list, this function removes the player and displays a message to the screen indicating that the player has been removed from the players list (list of lists). This function returns the list of players (list of lists). You may use the `list_name.append(item)` method in this function. You **must** call function `find_player()` from this function.

- **`play_blackjack_games(player_list, player_pos)`**

  This function allows a player to play Blackjack against the computer until he/she does not wish to continue playing (enters 'n' when prompted to 'Play again [y|n]?'). After every game, the player's chip balance and game stats are updated. 3 points are awarded if the player wins, 0 points are awarded if the player loses and 1 point is awarded if the player draws. The function takes the player list (list of lists) and the player's position (index) in the player list as parameters. This function calls function `play_one_game(no_chips)` (i.e. the function provided for you in the `blackjack` module; see note below). You **must** use a loop in your solution. **Please note: You DO NOT have to write your own game of Blackjack. One has been provided for you.** The `blackjack.py` file contains the function called `play_one_game(no_chips)` that allows you to play one game of Blackjack (dice version) against the computer. **Please do not modify this module.**

- **`sort_by_chips (player_list)`**

  This function takes the list of players (list of lists) and returns a **copy** of the list in descending order of chip balance. Where two players have the same chip balance, the player with the lower number of games played value should appear first. This function **must NOT** modify the player list being passed in as a parameter. Your solution **must NOT** make use of List methods or any of the sort functions available in the Python Standard Library. This function returns a **copy** of the player list (list of lists) sorted in descending order of chip balance. You may wish to create/define additional functions (which are called from within this function) to assist with this task.

- Good programming practice:
  - Consistent commenting, layout and indentation. You are to provide comments to describe: your details, program description, **all** variable definitions, **all** function definitions and every significant section of code.
  - Meaningful variable names.

Your solutions **MAY** make use of the following:

- Built-in functions `int()`, `input()`, `print()`, `range()`, `open()`, `close()`, `len()`, `format()` and `str()`.
- Concatenation (+) operator to create/build new strings.
- The `list_name.append(item)` method to update/create lists.
- Access the individual elements in a string with an index (one element only). i.e. `string_name[index]`.
- Access the individual elements in a list with an index (one element only). i.e. `list_name[index]` or `list_name[i][j]`.

Your solutions **MUST NOT** use:

- Built-in functions (other than the `int()`, `input()`, `print()`, `range()`, `open()`, `close()` `len()`, `format()` and `str()` functions).
- Slice expressions to select a range of elements from a string or list. i.e. `name[start:end]`.
- String or list methods (i.e. other than those mentioned in the 'MAY make use' of section above and those needed for the read_file() function).
- Global variables as described in week 10 material.
- The enumerate() function.
- List comprehensions.
- Dictionary to store data items.
- **Do not** use `break`, or `continue` statements in your solution – doing so will result in a significant mark deduction. **Do not** use the `return` statement as a way to break out of loops. **Do not** use `quit()` or `exit()` functions as a way to break out of loops.

PLEASE NOTE: You are reminded that you should ensure that all input and output conform to the assignment specifications. If you are not sure about these details you should check with the sample output provided at the end of this document or post a message to the discussion forum in order to seek clarification.

Please ensure that you use Python 3.12.2 or a later version (i.e. the latest version) in order to complete your assignments. Your programs **MUST** run using Python 3.12.2 (or latest version).

STAGES

It is recommended that you develop this part of the assignment in the suggested stages. Many problems in later stages are due to errors in early stages. **Make sure you have finished and thoroughly tested each stage before continuing.**

The following stages of development are recommended:

Stage 1
To begin, download the provided files (available on the course website – Assessment Tab) and re-name assign2_yourEmailId.py to *assign2_yourEmailId*.py. i.e. assign2_bonjy007.py

Define an empty list to store the player information. For example:

```
player_list = []
```

Call function play_one_game() (provided for you in the blackjack.py module). Note: **you may simply download it from the course website under the Assessment tab. The file blackjack.py should be placed in the same directory as your *assign2_yourEmailId*.py file.** Please do **NOT** modify the blackjack.py module.

Import the blackjack module and call function play_one_game() as seen below. You may wish to read the material on modules posted on the course website (under the assessment tab).

```
import blackjack


player_list = []

no_chips = 100
game_result = 0

# Plays one game of blackJack and assigns the result of the game and the chip balance
# to variables game_result and no_chips respectively.
game_result, no_chips = blackjack.play_one_game(no_chips)

# Display to the screen the result of play_one_game() -game result (value of 3, 1 or 0)
# and number of chips remaining.
print(game_result, no_chips)
```

Make sure the program runs correctly. Once you have that working, back up your program. *Note: When developing software, you should always have fixed points in your development where you know your software is bug free and runs correctly.*

Once you are sure that your program is working correctly, you can either remove or comment out the above statements – we just wanted to make sure it was working correctly! You can bring it back later (appropriately positioned).

Stage 2
Write the code for functions read_file() and display_players() as specified above. Add code to call these two functions to ensure they are working correctly. You may write your own read_file() function or you may use the one provided for you (included in the provided file).

Stage 3

Now that you know the information is being correctly stored in your players list, write the code for function `write_to_file()`. Add code to call this function to ensure it is working correctly.

Stage 4

Implement the interactive mode, i.e. to prompt for and read menu commands. Set up a loop to obtain and process commands. Test to ensure that this is working correctly before moving onto the next stage. You do not need to call any functions at this point, you may simply display an appropriate message to the screen, for example:

Sample output:

```
Please enter choice
[list, buy, search, high, add, remove, play, chips, quit]: display

Not a valid command - please try again.


Please enter choice
[list, buy, search, high, add, remove, play, chips, quit]: list

In list command


Please enter choice
[list, buy, search, high, add, remove, play, chips, quit]: buy

In buy command


Please enter choice
[list, buy, search, high, add, remove, play, chips, quit]: search

In search command


Please enter choice
[list, buy, search, high, add, remove, play, chips, quit]: high

In high command


Please enter choice
[list, buy, search, high, add, remove, play, chips, quit]: add

In add command


Please enter choice
[list, buy, search, high, add, remove, play, chips, quit]: remove

In remove command


Please enter choice
[list, buy, search, high, add, remove, play, chips, quit]: play

In play command


Please enter choice
[list, buy, search, high, add, remove, play, chips, quit]: chips

In chips command


Please enter choice
[list, buy, search, high, add, remove, play, chips, quit]: quit
```

Menu input should be validated with an appropriate message being displayed if incorrect input is entered by the user.

Stage 5

Implement one command at a time. Test to ensure the command is working correctly before starting the next command. Start with the `quit` and `list` commands as they do not need you to add anything further to the file other than ensuring that the function calls are in the correct place.

You should be able to see that for *most* commands there is a corresponding function(s).

For the remaining commands, the following implementation order is suggested (note: this is a guide only):

- `list` command (`display_players()` function).
- `search` command (`find_player()` function).
- `add` command (`add_player()` function).
- `buy` command (`buy_player_chips()` function).
- `remove` command (`remove_player()` function).
- `high` command (`display_highest_chip_holder()` function).
- `play` command (`play_blackjack_games()` function).
- `chips` command (`sort_by_chips()` function).

Stage 6

Ensure that you have validated all user input with an appropriate message being displayed for incorrect input entered by the user. Add code to validate all user input. Hint: use a while loop to validate input.

Stage 7

Finally, **check the sample output** (see section titled 'Sample Output' towards the end of this document) **and if necessary, modify your code so that:**

- The output produced by your program **EXACTLY** adheres to the sample output provided.
- Your program behaves as described in these specs and the sample output provided.

SUBMISSION DETAILS

You are required to do the following in order to submit your work and have it marked:

- You are required to submit an electronic copy of your program via learnonline **before Tuesday 11 June (swot-vac), 10.00 am**.

All students must follow the submission instructions below:

Ensure that your files are named correctly (as per instructions outlined in this document).

Ensure that the following file is included in your submission:

- `assign2_yourEmailId.py`

For example (if your name is James Bond, your submission file would be as follows):

- `assign2_bonjy007.py`

All files that you submit must include the following comments.

```
#
# File: fileName.py
# Author: your name
# Email Id: your email id
# Description: Assignment 2 - place assignment description here…
# This is my own work as defined by the University's
# Academic Misconduct policy.
#
```

Assignments that do not contain these details may not be marked.

You must submit your program **before the due date** via learnonline. Work that has not been correctly submitted to learnonline will not be marked.

It is expected that students will make copies of all assignments and be able to provide these if required.

EXTENSIONS AND LATE SUBMISSIONS

There will be **no** extensions/late submissions for this course without one of the following exceptions:

1. A medical certificate is provided that has the timing and duration of the illness and an opinion on how much the student's ability to perform has been compromised by the illness. **Please note** if this information is not provided the medical certificate WILL NOT BE ACCEPTED. Late assessment items will not be accepted unless a medical certificate is presented to the Course Coordinator. The certificate must be produced as soon as possible and must cover the dates during which the assessment was to be attempted. In the case where you have a valid medical certificate, the due date will be extended by the number of days stated on the certificate up to five working days.
2. A Learning and Teaching Unit councillor contacts the Course Coordinator on your behalf requesting an extension. Normally you would use this if you have events outside your control adversely affecting your course work.
3. Unexpected work commitments. In this case, you will need to attach a letter from your work supervisor with your application stating the impact on your ability to complete your assessment.
4. Military obligations with proof.

Applications for extensions must be lodged via learnonline before the due date of the assignment.

Note: Equipment failure, loss of data, 'Heavy work commitments' or late starting of the course are not sufficient grounds for an extension.

ACADEMIC MISCONDUCT

Students are reminded that they should be aware of what constitutes academic misconduct. Information about academic integrity and what constitutes academic misconduct can be found in the Academic Integrity Policy and Procedure (https://i.unisa.edu.au/policies-and-procedures/university-policies/academic/ab-69).

Deliberate academic misconduct (such as plagiarism, contract cheating, the use of ChatGPT (or similar tool)) is subject to penalties as outlined in the **Academic Integrity Policy and Procedure**.

You are **NOT** permitted to use ChatGPT (or other similar Artificial Intelligence tools) for your assessment items in this course. **Please note**: as stated in the University's Academic Integrity Policy (Policy AB-69), academic misconduct includes (but is not limited to) the use of artificial intelligence (AI) software or paraphrasing tools as a form of contract cheating. Please refer to the Academic Integrity Policy (Policy AB-69) for further information.

MARKING CRITERIA

Please note that the following is a guide only and may be subject to change (see next page for breakdown).

Other possible deductions:

- *Programming style:* Things to watch for are poor or no commenting, poor variable names, etc.
- *Submitted incorrectly:* -10 marks if assignment is submitted incorrectly (i.e. not adhering to the specs).

University of South Australia

Assessment feedback

COMP 1039 Problem Solving and Programming. Assignment Two – SP2, 2024

NAME:	AVAILABLE MARKS	MARK	COMMENT
PRODUCES CORRECT RESULTS (OUTPUT)	**60 MARKS**		
	☐ Line spacing (2 marks)		
File : assign2_wayby001.py Author : Batman Stud ID : 0123456X Email ID : wayby001 This is my own work as defined by the University's Academic Misconduct Policy.	2 Marks		
Please enter choice [list, buy, search, high, add, remove, play, chips, quit]:	2 Marks		
list Player Summary P W L D Chips Score Bruce Wayne 5 5 0 0 100 15 Jessica Jones 12 0 6 6 10 6 Johnny Rose 6 2 0 4 20 10 Gina Linetti 7 4 0 3 300 15 Buster Bluth 3 0 2 1 50 1	8 Marks -1 For each missing or incorrect info (up to 4 marks) -1 For each formatting error (up to 4 marks)		
buy Please enter name: Johnny Rose Johnny Rose currently has 20 chips. How many chips would you like to buy? 2000 You may only buy between 1-100 chips at a time! How many chips would you like to buy? -1 You may only buy between 1-100 chips at a time! How many chips would you like to buy? 80 Successfully updated Johnny Rose's chip balance to 100	3 Marks -3 If no/incorrect results -1 For each output/prompt/msg not to specs (up to 3 marks)		
search Please enter name: Gina Linetti Gina Linetti stats: P W L D Score 7 4 0 3 15 Chips: 300	5 Marks -5 If no/incorrect results -1 For each output/prompt/msg not to specs (up to 5 marks)		
high Highest Chip Holder => Gina Linetti with 300 chips!	5 Marks -5 If no/incorrect results -1 For each output/prompt/msg not to specs (up to 5 marks) *Check for equal chips.* *-2 if not using games played.*		
add Please enter name: Alexis Rose Successfully added Alexis Rose to player list.	5 Marks -5 If no/incorrect results -1 For each output/prompt/msg not to specs (up to 5 marks)		

remove
`Please enter name: Bruce Wayne`
`Successfully removed Bruce Wayne from player list.`
	5 Marks
	-5 If no/incorrect results
	-1 For each output/prompt/msg not to specs (up to 5 marks)		

play
`Please enter name: Buster Bluth`
`---------- START GAME ----------`
`:`
`:`
`:`
`---------- END GAME ----------`
`Play again [y|n]? n`
	8 Marks
	-8 If no/incorrect results
	-1 For each output/prompt/msg not to specs (up to 4 marks)
	Check to see whether stats and chips are updating correctly as a result of games. -2 if not updating correctly.		

chips
`Player Summary`
`Gina Linetti 7 4 0 3 300 15`
`Bruce Wayne 5 5 0 0 100 15`
`Buster Bluth 3 0 2 1 50 1`
`Johnny Rose 6 2 0 4 20 10`
`Jessica Jones 12 0 6 6 10 6`
	10 Marks
	-10 If no/incorrect results
	-1 For each output/prompt/msg not to specs (up to 5 marks)
	Check if equal chip balance -2 if not using games played.		

Output file (new_players.txt)	5 Marks -5 If output file does not exist -2 If incorrect results in file -2 If output not to specs in file -2 if not called new_players.txt		

ADHERES TO SPECIFICATIONS (CODE)		**COMMENT**
While/for loop for menu/prompt (choice != 'quit' or equivalent)		☐ -2 No or incorrect while
Use of List of Lists to store player information		☐ -2 No or incorrect use of lists
Use of supplied `blackjack.py` module and `play_one_game(no_chips)` function.		☐ -2 No or incorrect use of module and function
Functions: ▪ read_file(filename); returns list of players ▪ display_players(player_list) ▪ write_to_file(filename, player_list) ▪ find_player(player_list, name); return index or -1 ▪ add_player(player_list, name) return list of players; calls function find_player() ▪ remove_player(player_list, name); returns list of players; calls function find_player() ▪ display_highest_chip_holder(player_list) ▪ buy_player_chips(player_list, name) ▪ play_blackjack_games(player_list, player_pos) ▪ sort_by_chips(player_list); returns copy of player list sorted in descending order of chips		☐ -2 No or incorrect read function ☐ -2 No or incorrect display function ☐ -2 No or incorrect write function ☐ -2 No or incorrect find function ☐ -2 No or incorrect add function ☐ -2 No or incorrect remove function ☐ -2 No or incorrect highest function ☐ -2 No or incorrect buy function ☐ -2 No or incorrect play function ☐ -2 No or incorrect sort function

Should **not use** the following: ▪ Built-in functions (other than those allowed), ▪ Slice expressions to select range of elements, ▪ String or list methods (other than list_name.append() and those allowed in read_file()). ▪ The enumerate() function. ▪ List comprehensions. ▪ Dictionary to store data items.			☐ -2 For use of bult-ins not allowed ☐ -2 For use of slice expressions ☐ -2 For use of methods not allowed ☐ -10 use of enumerate() ☐ -10 use of list comprehensions ☐ -10 use of dictionary
No global variables			☐ -2 For use of global variables
No magic numbers (use constant variables for numbers)			☐ -2 For use of magic numbers
Validation of input: `Not a valid command - please try again.`			☐ -2 No validation of command
Good loops (i.e. no break, continue, return or similar statements to exit loops)			☐ -2 For using break/return/quit/etc statements to exit loops
STYLE	**5 MARKS**	**MARK**	**COMMENT**
Comments: ○ your details at top of file ○ program description ○ all variable definitions ○ all function definitions ○ significant sections of code			☐ -4 Insufficient comments ○ your details at top of file, ○ program description, ○ all variable definitions, ○ functions, and ○ significant sections of code
Consistent code layout and indentation.			☐ -4 Inconsistent indentation and layout
Meaningful variable names (no single letter variable names).			☐ -4 Non-descriptive variable names
TOTAL	**65 MARKS**		

The Graduate qualities being assessed by this assignment are indicated by an X:

X	GQ1: operate effectively with and upon a body of knowledge		GQ5: are committed to ethical action and social responsibility
	GQ2: are prepared for lifelong learning	X	GQ6: communicate effectively
	GQ3: are effective problem solvers		GQ7: demonstrate an international perspective
	GQ4:can work both autonomously and collaboratively		

This form meets the 2006 requirements of UniSA's Code of Good Practice: Student Assessment

SAMPLE OUTPUT

Sample output 1:

```
File     : assign2_wayby001.py
Author   : Batman
Stud ID  : 0123456X
Email ID : wayby001
This is my own work as defined by the University's Academic Misconduct Policy.


Please enter choice
[list, buy, search, high, add, remove, play, chips, quit]: list

============================================================
-                     Player Summary                       -
============================================================
-                             P  W  L  D   Chips   Score  -
------------------------------------------------------------
-  Bruce Wayne                5  5  0  0     100      15  -
------------------------------------------------------------
-  Jessica Jones             12  0  6  6      10       6  -
------------------------------------------------------------
-  Johnny Rose                6  2  0  4      20      10  -
------------------------------------------------------------
-  Gina Linetti               7  4  0  3     300      15  -
------------------------------------------------------------
-  Buster Bluth               3  0  2  1      50       1  -
------------------------------------------------------------
============================================================

Please enter choice
[list, buy, search, high, add, remove, play, chips, quit]: display

Not a valid command - please try again.


Please enter choice
[list, buy, search, high, add, remove, play, chips, quit]: quit


-- Program terminating --


NOTE: Your program should output the following information to a file - new_players.txt.

Bruce Wayne
5 5 0 0 100 15
Jessica Jones
12 0 6 6 10 6
Johnny Rose
6 2 0 4 20 10
Gina Linetti
7 4 0 3 300 15
Buster Bluth
3 0 2 1 50 1
```

Sample output 2:

```
File     : assign2_wayby001.py
Author   : Batman
Stud ID  : 0123456X
Email ID : wayby001
This is my own work as defined by the University's Academic Misconduct Policy.


Please enter choice
[list, buy, search, high, add, remove, play, chips, quit]: list

============================================================
-                     Player Summary                       -
============================================================
-                             P  W  L  D   Chips   Score  -
------------------------------------------------------------
-  Bruce Wayne                5  5  0  0     100      15  -
------------------------------------------------------------
-  Jessica Jones             12  0  6  6      10       6  -
------------------------------------------------------------
-  Johnny Rose                6  2  0  4      20      10  -
------------------------------------------------------------
-  Gina Linetti               7  4  0  3     300      15  -
------------------------------------------------------------
-  Buster Bluth               3  0  2  1      50       1  -
------------------------------------------------------------
============================================================

Please enter choice
[list, buy, search, high, add, remove, play, chips, quit]: buy

Please enter name: Tony Stark
```

```
Tony Stark is not found in player list.


Please enter choice
[list, buy, search, high, add, remove, play, chips, quit]: buy

Please enter name: Johnny Rose

Johnny Rose currently has 20 chips.

How many chips would you like to buy? 2000
You may only buy between 1-100 chips at a time!

How many chips would you like to buy? -1
You may only buy between 1-100 chips at a time!

How many chips would you like to buy? 80

Successfully updated Johnny Rose's chip balance to 100

Please enter choice
[list, buy, search, high, add, remove, play, chips, quit]: list

============================================================
-                     Player Summary                       -
============================================================
-                              P  W  L  D   Chips   Score  -
------------------------------------------------------------
-  Bruce Wayne                 5  5  0  0     100      15  -
------------------------------------------------------------
-  Jessica Jones              12  0  6  6      10       6  -
------------------------------------------------------------
-  Johnny Rose                 6  2  0  4     100      10  -
------------------------------------------------------------
-  Gina Linetti                7  4  0  3     300      15  -
------------------------------------------------------------
-  Buster Bluth                3  0  2  1      50       1  -
------------------------------------------------------------
============================================================

Please enter choice
[list, buy, search, high, add, remove, play, chips, quit]: quit


-- Program terminating --


NOTE: Your program should output the following information to a file - new_players.txt.

Bruce Wayne
5 5 0 0 100 15
Jessica Jones
12 0 6 6 10 6
Johnny Rose
6 2 0 4 100 10
Gina Linetti
7 4 0 3 300 15
Buster Bluth
3 0 2 1 50 1
```

Sample output 3:

```
File     : assign2_wayby001.py
Author   : Batman
Stud ID  : 0123456X
Email ID : wayby001
This is my own work as defined by the University's Academic Misconduct Policy.


Please enter choice
[list, buy, search, high, add, remove, play, chips, quit]: list

============================================================
-                     Player Summary                       -
============================================================
-                              P  W  L  D   Chips   Score  -
------------------------------------------------------------
-  Bruce Wayne                 5  5  0  0     100      15  -
------------------------------------------------------------
-  Jessica Jones              12  0  6  6      10       6  -
------------------------------------------------------------
-  Johnny Rose                 6  2  0  4      20      10  -
------------------------------------------------------------
-  Gina Linetti                7  4  0  3     300      15  -
------------------------------------------------------------
-  Buster Bluth                3  0  2  1      50       1  -
------------------------------------------------------------
============================================================

Please enter choice
[list, buy, search, high, add, remove, play, chips, quit]: search

Please enter name: Thomas Anderson
```

```
Thomas Anderson is not found in player list.


Please enter choice
[list, buy, search, high, add, remove, play, chips, quit]: search

Please enter name: Gina Linetti

Gina Linetti stats:

P  W  L  D  Score
7  4  0  3  15

Chips:  300


Please enter choice
[list, buy, search, high, add, remove, play, chips, quit]: quit


-- Program terminating --


NOTE: Your program should output the following information to a file - new_players.txt.

Bruce Wayne
5 5 0 0 100 15
Jessica Jones
12 0 6 6 10 6
Johnny Rose
6 2 0 4 20 10
Gina Linetti
7 4 0 3 300 15
Buster Bluth
3 0 2 1 50 1
```

Sample output 4:

```
File     : assign2_wayby001.py
Author   : Batman
Stud ID  : 0123456X
Email ID : wayby001
This is my own work as defined by the University's Academic Misconduct Policy.


Please enter choice
[list, buy, search, high, add, remove, play, chips, quit]: list

============================================================
-                    Player Summary                        -
============================================================
-                            P  W  L  D   Chips   Score  -
------------------------------------------------------------
-  Bruce Wayne               5  5  0  0     100      15  -
------------------------------------------------------------
-  Jessica Jones            12  0  6  6      10       6  -
------------------------------------------------------------
-  Johnny Rose               6  2  0  4      20      10  -
------------------------------------------------------------
-  Gina Linetti              7  4  0  3     300      15  -
------------------------------------------------------------
-  Buster Bluth              3  0  2  1      50       1  -
------------------------------------------------------------
============================================================

Please enter choice
[list, buy, search, high, add, remove, play, chips, quit]: high

Highest Chip Holder => Gina Linetti with 300 chips!


Please enter choice
[list, buy, search, high, add, remove, play, chips, quit]: quit


-- Program terminating --


NOTE: Your program should output the following information to a file - new_players.txt.

Bruce Wayne
5 5 0 0 100 15
Jessica Jones
12 0 6 6 10 6
Johnny Rose
6 2 0 4 20 10
Gina Linetti
7 4 0 3 300 15
Buster Bluth
3 0 2 1 50 1
```

Sample output 5:

```
File     : assign2_wayby001.py
Author   : Batman
```

```
Stud ID  : 0123456X
Email ID : wayby001
This is my own work as defined by the University's Academic Misconduct Policy.


Please enter choice
[list, buy, search, high, add, remove, play, chips, quit]: list

============================================================
-                     Player Summary                       -
============================================================
-                             P  W  L  D   Chips   Score  -
------------------------------------------------------------
-  Bruce Wayne                5  5  0  0     100      15  -
------------------------------------------------------------
-  Jessica Jones             12  0  6  6      10       6  -
------------------------------------------------------------
-  Johnny Rose                6  2  0  4      20      10  -
------------------------------------------------------------
-  Gina Linetti               7  4  0  3     300      15  -
------------------------------------------------------------
-  Buster Bluth               3  0  2  1      50       1  -
------------------------------------------------------------
============================================================

Please enter choice
[list, buy, search, high, add, remove, play, chips, quit]: add

Please enter name: Jessica Jones

Jessica Jones already exists in player list.


Please enter choice
[list, buy, search, high, add, remove, play, chips, quit]: add

Please enter name: Alexis Rose

Successfully added Alexis Rose to player list.


Please enter choice
[list, buy, search, high, add, remove, play, chips, quit]: list

============================================================
-                     Player Summary                       -
============================================================
-                             P  W  L  D   Chips   Score  -
------------------------------------------------------------
-  Bruce Wayne                5  5  0  0     100      15  -
------------------------------------------------------------
-  Jessica Jones             12  0  6  6      10       6  -
------------------------------------------------------------
-  Johnny Rose                6  2  0  4      20      10  -
------------------------------------------------------------
-  Gina Linetti               7  4  0  3     300      15  -
------------------------------------------------------------
-  Buster Bluth               3  0  2  1      50       1  -
------------------------------------------------------------
-  Alexis Rose                0  0  0  0     100       0  -
------------------------------------------------------------
============================================================

Please enter choice
[list, buy, search, high, add, remove, play, chips, quit]: quit


-- Program terminating --


NOTE: Your program should output the following information to a file - new_players.txt.

Bruce Wayne
5 5 0 0 100 15
Jessica Jones
12 0 6 6 10 6
Johnny Rose
6 2 0 4 20 10
Gina Linetti
7 4 0 3 300 15
Buster Bluth
3 0 2 1 50 1
Alexis Rose
0 0 0 0 100 0
```

Sample output 6:

```
File     : assign2_wayby001.py
Author   : Batman
Stud ID  : 0123456X
Email ID : wayby001
This is my own work as defined by the University's Academic Misconduct Policy.
```

```
Please enter choice
[list, buy, search, high, add, remove, play, chips, quit]: list

============================================================
-                     Player Summary                       -
============================================================
-                             P  W  L  D   Chips   Score  -
------------------------------------------------------------
-  Bruce Wayne                5  5  0  0     100      15  -
------------------------------------------------------------
-  Jessica Jones             12  0  6  6      10       6  -
------------------------------------------------------------
-  Johnny Rose                6  2  0  4      20      10  -
------------------------------------------------------------
-  Gina Linetti               7  4  0  3     300      15  -
------------------------------------------------------------
-  Buster Bluth               3  0  2  1      50       1  -
------------------------------------------------------------
============================================================

Please enter choice
[list, buy, search, high, add, remove, play, chips, quit]: remove

Please enter name: Natasha Romanoff

Natasha Romanoff is not found in players.


Please enter choice
[list, buy, search, high, add, remove, play, chips, quit]: remove

Please enter name: Bruce Wayne

Successfully removed Bruce Wayne from player list.


Please enter choice
[list, buy, search, high, add, remove, play, chips, quit]: list

============================================================
-                     Player Summary                       -
============================================================
-                             P  W  L  D   Chips   Score  -
------------------------------------------------------------
-  Jessica Jones             12  0  6  6      10       6  -
------------------------------------------------------------
-  Johnny Rose                6  2  0  4      20      10  -
------------------------------------------------------------
-  Gina Linetti               7  4  0  3     300      15  -
------------------------------------------------------------
-  Buster Bluth               3  0  2  1      50       1  -
------------------------------------------------------------
============================================================

Please enter choice
[list, buy, search, high, add, remove, play, chips, quit]: remove

Please enter name: Johnny Rose

Successfully removed Johnny Rose from player list.


Please enter choice
[list, buy, search, high, add, remove, play, chips, quit]: list

============================================================
-                     Player Summary                       -
============================================================
-                             P  W  L  D   Chips   Score  -
------------------------------------------------------------
-  Jessica Jones             12  0  6  6      10       6  -
------------------------------------------------------------
-  Gina Linetti               7  4  0  3     300      15  -
------------------------------------------------------------
-  Buster Bluth               3  0  2  1      50       1  -
------------------------------------------------------------
============================================================

Please enter choice
[list, buy, search, high, add, remove, play, chips, quit]: quit


-- Program terminating --


NOTE: Your program should output the following information to a file - new_players.txt.

Jessica Jones
12 0 6 6 10 6
Gina Linetti
7 4 0 3 300 15
Buster Bluth
3 0 2 1 50 1
```

Sample output 7:

```
File     : assign2_wayby001.py
Author   : Batman
Stud ID  : 0123456X
Email ID : wayby001
This is my own work as defined by the University's Academic Misconduct Policy.


Please enter choice
[list, buy, search, high, add, remove, play, chips, quit]: list

============================================================
-                     Player Summary                       -
============================================================
-                             P  W  L  D   Chips   Score  -
------------------------------------------------------------
-  Bruce Wayne                5  5  0  0     100      15  -
------------------------------------------------------------
-  Jessica Jones             12  0  6  6      10       6  -
------------------------------------------------------------
-  Johnny Rose                6  2  0  4      20      10  -
------------------------------------------------------------
-  Gina Linetti               7  4  0  3     300      15  -
------------------------------------------------------------
-  Buster Bluth               3  0  2  1      50       1  -
------------------------------------------------------------
============================================================

Please enter choice
[list, buy, search, high, add, remove, play, chips, quit]: play

Please enter name: Harold Crick

Harold Crick is not found in player list.


Please enter choice
[list, buy, search, high, add, remove, play, chips, quit]: play

Please enter name: Buster Bluth


---------------------- START GAME ----------------------
| Dealer's hand is: 7
| Player's hand is: 5
|
| --> Number of chips: 50
| --> Place your bet:  50
|
| Player's hand is: 5 + 10 = 15
| Please enter h or s (h = Hit, s = Stand): h
| Player's hand is: 15 + 7 = 22
| *** Player busts!
|
| Dealer's hand is: 7 + 10 = 17
|
| *** Dealer: 17  Player: 22  -  Dealer Wins! ***
| --> 0 chips left!
----------------------- END GAME -----------------------


Play again [y|n]? n

Please enter choice
[list, buy, search, high, add, remove, play, chips, quit]: list

============================================================
-                     Player Summary                       -
============================================================
-                             P  W  L  D   Chips   Score  -
------------------------------------------------------------
-  Bruce Wayne                5  5  0  0     100      15  -
------------------------------------------------------------
-  Jessica Jones             12  0  6  6      10       6  -
------------------------------------------------------------
-  Johnny Rose                6  2  0  4      20      10  -
------------------------------------------------------------
-  Gina Linetti               7  4  0  3     300      15  -
------------------------------------------------------------
-  Buster Bluth               4  0  3  1       0       1  -
------------------------------------------------------------
============================================================

Please enter choice
[list, buy, search, high, add, remove, play, chips, quit]: buy

Please enter name: Buster Bluth

Buster Bluth currently has 0 chips.

How many chips would you like to buy? 100

Successfully updated Buster Bluth's chip balance to 100
```

```
Please enter choice
[list, buy, search, high, add, remove, play, chips, quit]: list

============================================================
-                     Player Summary                       -
============================================================
-                             P  W  L  D   Chips   Score  -
------------------------------------------------------------
-  Bruce Wayne                5  5  0  0     100      15  -
------------------------------------------------------------
-  Jessica Jones             12  0  6  6      10       6  -
------------------------------------------------------------
-  Johnny Rose                6  2  0  4      20      10  -
------------------------------------------------------------
-  Gina Linetti               7  4  0  3     300      15  -
------------------------------------------------------------
-  Buster Bluth               4  0  3  1     100       1  -
------------------------------------------------------------
============================================================

Please enter choice
[list, buy, search, high, add, remove, play, chips, quit]: play

Please enter name: Buster Bluth


---------------------- START GAME ----------------------
| Dealer's hand is: 8
| Player's hand is: 9
|
| --> Number of chips: 100
| --> Place your bet:  50
|
| Player's hand is: 9 + 9 = 18
| Please enter h or s (h = Hit, s = Stand): s
|
| Dealer's hand is: 8 + 11 = 19
|
| *** Dealer: 19  Player: 18  -  Dealer Wins! ***
| --> 50 chips left!
---------------------- END GAME ----------------------


Play again [y|n]? y


---------------------- START GAME ----------------------
| Dealer's hand is: 6
| Player's hand is: 6
|
| --> Number of chips: 50
| --> Place your bet:  20
|
| Player's hand is: 6 + 6 = 12
| Please enter h or s (h = Hit, s = Stand): h
| Player's hand is: 12 + 7 = 19
| Please enter h or s (h = Hit, s = Stand): s
|
| Dealer's hand is: 6 + 6 = 12
| Dealer's hand is: 12 + 6 = 18
|
| *** Dealer: 18  Player: 19  -  Player Wins! ***
| --> 70 chips left!
---------------------- END GAME ----------------------


Play again [y|n]? n

Please enter choice
[list, buy, search, high, add, remove, play, chips, quit]: list

============================================================
-                     Player Summary                       -
============================================================
-                             P  W  L  D   Chips   Score  -
------------------------------------------------------------
-  Bruce Wayne                5  5  0  0     100      15  -
------------------------------------------------------------
-  Jessica Jones             12  0  6  6      10       6  -
------------------------------------------------------------
-  Johnny Rose                6  2  0  4      20      10  -
------------------------------------------------------------
-  Gina Linetti               7  4  0  3     300      15  -
------------------------------------------------------------
-  Buster Bluth               6  1  4  1      70       4  -
------------------------------------------------------------
============================================================

Please enter choice
[list, buy, search, high, add, remove, play, chips, quit]: quit


-- Program terminating --
```

```
NOTE: Your program should output the following information to a file - new_players.txt.

Bruce Wayne
5 5 0 0 100 15
Jessica Jones
12 0 6 6 10 6
Johnny Rose
6 2 0 4 20 10
Gina Linetti
7 4 0 3 300 15
Buster Bluth
6 1 4 1 70 4
```

Sample output 8:

```
File     : assign2_wayby001.py
Author   : Batman
Stud ID  : 0123456X
Email ID : wayby001
This is my own work as defined by the University's Academic Misconduct Policy.


Please enter choice
[list, buy, search, high, add, remove, play, chips, quit]: list

============================================================
-                     Player Summary                       -
============================================================
-                            P  W  L  D   Chips   Score  -
------------------------------------------------------------
-  Bruce Wayne               5  5  0  0     100      15  -
------------------------------------------------------------
-  Jessica Jones            12  0  6  6      10       6  -
------------------------------------------------------------
-  Johnny Rose               6  2  0  4      20      10  -
------------------------------------------------------------
-  Gina Linetti              7  4  0  3     300      15  -
------------------------------------------------------------
-  Buster Bluth              3  0  2  1      50       1  -
------------------------------------------------------------
============================================================

Please enter choice
[list, buy, search, high, add, remove, play, chips, quit]: chips

============================================================
-                     Player Summary                       -
============================================================
-                            P  W  L  D   Chips   Score  -
------------------------------------------------------------
-  Gina Linetti              7  4  0  3     300      15  -
------------------------------------------------------------
-  Bruce Wayne               5  5  0  0     100      15  -
------------------------------------------------------------
-  Buster Bluth              3  0  2  1      50       1  -
------------------------------------------------------------
-  Johnny Rose               6  2  0  4      20      10  -
------------------------------------------------------------
-  Jessica Jones            12  0  6  6      10       6  -
------------------------------------------------------------
============================================================

Please enter choice
[list, buy, search, high, add, remove, play, chips, quit]: list

============================================================
-                     Player Summary                       -
============================================================
-                            P  W  L  D   Chips   Score  -
------------------------------------------------------------
-  Bruce Wayne               5  5  0  0     100      15  -
------------------------------------------------------------
-  Jessica Jones            12  0  6  6      10       6  -
------------------------------------------------------------
-  Johnny Rose               6  2  0  4      20      10  -
------------------------------------------------------------
-  Gina Linetti              7  4  0  3     300      15  -
------------------------------------------------------------
-  Buster Bluth              3  0  2  1      50       1  -
------------------------------------------------------------
============================================================

Please enter choice
[list, buy, search, high, add, remove, play, chips, quit]: buy

Please enter name: Buster Bluth

Buster Bluth currently has 50 chips.

How many chips would you like to buy? 50

Successfully updated Buster Bluth's chip balance to 100
```

```
Please enter choice
[list, buy, search, high, add, remove, play, chips, quit]: list

============================================================
-                     Player Summary                       -
============================================================
-                             P  W  L  D   Chips   Score  -
------------------------------------------------------------
-  Bruce Wayne                5  5  0  0     100      15  -
------------------------------------------------------------
-  Jessica Jones             12  0  6  6      10       6  -
------------------------------------------------------------
-  Johnny Rose                6  2  0  4      20      10  -
------------------------------------------------------------
-  Gina Linetti               7  4  0  3     300      15  -
------------------------------------------------------------
-  Buster Bluth               3  0  2  1     100       1  -
------------------------------------------------------------
============================================================

Please enter choice
[list, buy, search, high, add, remove, play, chips, quit]: chips

============================================================
-                     Player Summary                       -
============================================================
-                             P  W  L  D   Chips   Score  -
------------------------------------------------------------
-  Gina Linetti               7  4  0  3     300      15  -
------------------------------------------------------------
-  Buster Bluth               3  0  2  1     100       1  -
------------------------------------------------------------
-  Bruce Wayne                5  5  0  0     100      15  -
------------------------------------------------------------
-  Johnny Rose                6  2  0  4      20      10  -
------------------------------------------------------------
-  Jessica Jones             12  0  6  6      10       6  -
------------------------------------------------------------
============================================================

Please enter choice
[list, buy, search, high, add, remove, play, chips, quit]: quit


-- Program terminating --


NOTE: Your program should output the following information to a file - new_players.txt.

Bruce Wayne
5 5 0 0 100 15
Jessica Jones
12 0 6 6 10 6
Johnny Rose
6 2 0 4 20 10
Gina Linetti
7 4 0 3 300 15
Buster Bluth
3 0 2 1 100 1
```

Sample output 9:

```
File     : assign2_wayby001.py
Author   : Batman
Stud ID  : 0123456X
Email ID : wayby001
This is my own work as defined by the University's Academic Misconduct Policy.


Please enter choice
[list, buy, search, high, add, remove, play, chips, quit]: list

============================================================
-                     Player Summary                       -
============================================================
-                             P  W  L  D   Chips   Score  -
------------------------------------------------------------
-  Bruce Wayne                5  5  0  0     100      15  -
------------------------------------------------------------
-  Jessica Jones             12  0  6  6      10       6  -
------------------------------------------------------------
-  Johnny Rose                6  2  0  4      20      10  -
------------------------------------------------------------
-  Gina Linetti               7  4  0  3     300      15  -
------------------------------------------------------------
-  Buster Bluth               3  0  2  1      50       1  -
------------------------------------------------------------
============================================================

Please enter choice
[list, buy, search, high, add, remove, play, chips, quit]: add

Please enter name: Bruce Banner
```

```
Successfully added Bruce Banner to player list.


Please enter choice
[list, buy, search, high, add, remove, play, chips, quit]: list

============================================================
-                      Player Summary                      -
============================================================
-                             P  W  L  D   Chips   Score  -
------------------------------------------------------------
-  Bruce Wayne                5  5  0  0     100      15  -
------------------------------------------------------------
-  Jessica Jones             12  0  6  6      10       6  -
------------------------------------------------------------
-  Johnny Rose                6  2  0  4      20      10  -
------------------------------------------------------------
-  Gina Linetti               7  4  0  3     300      15  -
------------------------------------------------------------
-  Buster Bluth               3  0  2  1      50       1  -
------------------------------------------------------------
-  Bruce Banner               0  0  0  0     100       0  -
------------------------------------------------------------
============================================================

Please enter choice
[list, buy, search, high, add, remove, play, chips, quit]: add

Please enter name: Bruce Banner

Bruce Banner already exists in player list.


Please enter choice
[list, buy, search, high, add, remove, play, chips, quit]: list

============================================================
-                      Player Summary                      -
============================================================
-                             P  W  L  D   Chips   Score  -
------------------------------------------------------------
-  Bruce Wayne                5  5  0  0     100      15  -
------------------------------------------------------------
-  Jessica Jones             12  0  6  6      10       6  -
------------------------------------------------------------
-  Johnny Rose                6  2  0  4      20      10  -
------------------------------------------------------------
-  Gina Linetti               7  4  0  3     300      15  -
------------------------------------------------------------
-  Buster Bluth               3  0  2  1      50       1  -
------------------------------------------------------------
-  Bruce Banner               0  0  0  0     100       0  -
------------------------------------------------------------
============================================================

Please enter choice
[list, buy, search, high, add, remove, play, chips, quit]: play

Please enter name: Bruce Banner


---------------------- START GAME ----------------------
| Dealer's hand is: 11
| Player's hand is: 4
|
| --> Number of chips: 100
| --> Place your bet:  10
|
| Player's hand is: 4 + 9 = 13
| Please enter h or s (h = Hit, s = Stand): h
| Player's hand is: 13 + 10 = 23
| *** Player busts!
|
| Dealer's hand is: 11 + 9 = 20
|
| *** Dealer: 20  Player: 23  -  Dealer Wins! ***
| --> 90 chips left!
---------------------- END GAME ----------------------


Play again [y|n]? y


---------------------- START GAME ----------------------
| Dealer's hand is: 11
| Player's hand is: 1
|
| --> Number of chips: 90
| --> Place your bet:  10
|
| Player's hand is: 1 + 2 = 3
| Please enter h or s (h = Hit, s = Stand): h
| Player's hand is: 3 + 10 = 13
```

```
| Please enter h or s (h = Hit, s = Stand): h
| Player's hand is: 13 + 8 = 21
|
| Dealer's hand is: 11 + 5 = 16
| Dealer's hand is: 16 + 7 = 23
| *** Dealer busts!
|
| *** Dealer: 23  Player: 21  -  Player Wins! ***
| --> 100 chips left!
---------------------- END GAME ----------------------


Play again [y|n]? y


---------------------- START GAME ----------------------
| Dealer's hand is: 6
| Player's hand is: 3
|
| --> Number of chips: 100
| --> Place your bet:  20
|
| Player's hand is: 3 + 8 = 11
| Please enter h or s (h = Hit, s = Stand): h
| Player's hand is: 11 + 6 = 17
| Please enter h or s (h = Hit, s = Stand): h
| Player's hand is: 17 + 3 = 20
| Please enter h or s (h = Hit, s = Stand): s
|
| Dealer's hand is: 6 + 1 = 7
| Dealer's hand is: 7 + 6 = 13
| Dealer's hand is: 13 + 6 = 19
|
| *** Dealer: 19  Player: 20  -  Player Wins! ***
| --> 120 chips left!
---------------------- END GAME ----------------------


Play again [y|n]? y


---------------------- START GAME ----------------------
| Dealer's hand is: 5
| Player's hand is: 4
|
| --> Number of chips: 120
| --> Place your bet:  50
|
| Player's hand is: 4 + 10 = 14
| Please enter h or s (h = Hit, s = Stand): h
| Player's hand is: 14 + 11 = 25
| *** Player busts!
|
| Dealer's hand is: 5 + 10 = 15
| Dealer's hand is: 15 + 3 = 18
|
| *** Dealer: 18  Player: 25  -  Dealer Wins! ***
| --> 70 chips left!
---------------------- END GAME ----------------------


Play again [y|n]? y


---------------------- START GAME ----------------------
| Dealer's hand is: 4
| Player's hand is: 7
|
| --> Number of chips: 70
| --> Place your bet:  70
|
| Player's hand is: 7 + 1 = 8
| Please enter h or s (h = Hit, s = Stand): h
| Player's hand is: 8 + 5 = 13
| Please enter h or s (h = Hit, s = Stand): h
| Player's hand is: 13 + 6 = 19
| Please enter h or s (h = Hit, s = Stand): s
|
| Dealer's hand is: 4 + 4 = 8
| Dealer's hand is: 8 + 1 = 9
| Dealer's hand is: 9 + 7 = 16
| Dealer's hand is: 16 + 10 = 26
| *** Dealer busts!
|
| *** Dealer: 26  Player: 19  -  Player Wins! ***
| --> 140 chips left!
---------------------- END GAME ----------------------


Play again [y|n]? n

Please enter choice
[list, buy, search, high, add, remove, play, chips, quit]: list
```

```
============================================================
-                     Player Summary                       -
============================================================
-                             P  W  L  D   Chips   Score  -
------------------------------------------------------------
-  Bruce Wayne                5  5  0  0     100      15  -
------------------------------------------------------------
-  Jessica Jones             12  0  6  6      10       6  -
------------------------------------------------------------
-  Johnny Rose                6  2  0  4      20      10  -
------------------------------------------------------------
-  Gina Linetti               7  4  0  3     300      15  -
------------------------------------------------------------
-  Buster Bluth               3  0  2  1      50       1  -
------------------------------------------------------------
-  Bruce Banner               5  3  2  0     140       9  -
------------------------------------------------------------
============================================================

Please enter choice
[list, buy, search, high, add, remove, play, chips, quit]: chips

============================================================
-                     Player Summary                       -
============================================================
-                             P  W  L  D   Chips   Score  -
------------------------------------------------------------
-  Gina Linetti               7  4  0  3     300      15  -
------------------------------------------------------------
-  Bruce Banner               5  3  2  0     140       9  -
------------------------------------------------------------
-  Bruce Wayne                5  5  0  0     100      15  -
------------------------------------------------------------
-  Buster Bluth               3  0  2  1      50       1  -
------------------------------------------------------------
-  Johnny Rose                6  2  0  4      20      10  -
------------------------------------------------------------
-  Jessica Jones             12  0  6  6      10       6  -
------------------------------------------------------------
============================================================

Please enter choice
[list, buy, search, high, add, remove, play, chips, quit]: list

============================================================
-                     Player Summary                       -
============================================================
-                             P  W  L  D   Chips   Score  -
------------------------------------------------------------
-  Bruce Wayne                5  5  0  0     100      15  -
------------------------------------------------------------
-  Jessica Jones             12  0  6  6      10       6  -
------------------------------------------------------------
-  Johnny Rose                6  2  0  4      20      10  -
------------------------------------------------------------
-  Gina Linetti               7  4  0  3     300      15  -
------------------------------------------------------------
-  Buster Bluth               3  0  2  1      50       1  -
------------------------------------------------------------
-  Bruce Banner               5  3  2  0     140       9  -
------------------------------------------------------------
============================================================

Please enter choice
[list, buy, search, high, add, remove, play, chips, quit]: quit


-- Program terminating --


NOTE: Your program should output the following information to a file - new_players.txt.

Bruce Wayne
5 5 0 0 100 15
Jessica Jones
12 0 6 6 10 6
Johnny Rose
6 2 0 4 20 10
Gina Linetti
7 4 0 3 300 15
Buster Bluth
3 0 2 1 50 1
Bruce Banner
5 3 2 0 140 9
```

Sample output 10:

```
File     : assign2_wayby001.py
Author   : Batman
Stud ID  : 0123456X
Email ID : wayby001
This is my own work as defined by the University's Academic Misconduct Policy.
```

```
Please enter choice
[list, buy, search, high, add, remove, play, chips, quit]: high

Highest Chip Holder => Gina Linetti with 300 chips!


Please enter choice
[list, buy, search, high, add, remove, play, chips, quit]: list

============================================================
-                     Player Summary                       -
============================================================
-                              P  W  L  D   Chips   Score  -
------------------------------------------------------------
-  Bruce Wayne                 5  5  0  0     100      15  -
------------------------------------------------------------
-  Jessica Jones              12  0  6  6      10       6  -
------------------------------------------------------------
-  Johnny Rose                 6  2  0  4      20      10  -
------------------------------------------------------------
-  Gina Linetti                7  4  0  3     300      15  -
------------------------------------------------------------
-  Buster Bluth                3  0  2  1      50       1  -
------------------------------------------------------------
============================================================

Please enter choice
[list, buy, search, high, add, remove, play, chips, quit]: buy

Please enter name: Bruce Wayne

Bruce Wayne currently has 100 chips.

How many chips would you like to buy? 100

Successfully updated Bruce Wayne's chip balance to 200

Please enter choice
[list, buy, search, high, add, remove, play, chips, quit]: buy

Please enter name: Bruce Wayne

Bruce Wayne currently has 200 chips.

How many chips would you like to buy? 100

Successfully updated Bruce Wayne's chip balance to 300

Please enter choice
[list, buy, search, high, add, remove, play, chips, quit]: list

============================================================
-                     Player Summary                       -
============================================================
-                              P  W  L  D   Chips   Score  -
------------------------------------------------------------
-  Bruce Wayne                 5  5  0  0     300      15  -
------------------------------------------------------------
-  Jessica Jones              12  0  6  6      10       6  -
------------------------------------------------------------
-  Johnny Rose                 6  2  0  4      20      10  -
------------------------------------------------------------
-  Gina Linetti                7  4  0  3     300      15  -
------------------------------------------------------------
-  Buster Bluth                3  0  2  1      50       1  -
------------------------------------------------------------
============================================================

Please enter choice
[list, buy, search, high, add, remove, play, chips, quit]: high

Highest Chip Holder => Bruce Wayne with 300 chips!


Please enter choice
[list, buy, search, high, add, remove, play, chips, quit]: quit


-- Program terminating --


NOTE: Your program should output the following information to a file - new_players.txt.

Bruce Wayne
5 5 0 0 300 15
Jessica Jones
12 0 6 6 10 6
Johnny Rose
6 2 0 4 20 10
Gina Linetti
7 4 0 3 300 15
Buster Bluth
3 0 2 1 50 1
```

Formatting Integers:
You can use the `format()` function to format the way integers and strings are displayed to the screen.
In the following example:

```
print(format(total_user_score, '10d'))
```

the integer value assigned to variable `total_user_score` is printed in a field that is 10 spaces wide. By default, it is right-aligned within the field width.

There are other alignment options as follows:

Option	Meaning
'<'	Forces the field to be left-aligned within the available space (this is the default for most objects).
'>'	Forces the field to be right-aligned within the available space (this is the default for numbers).
'^'	Forces the field to be centered within the available space.

More examples of use (including output):

```
>>> total_user_score = 100
>>> format(total_user_score, '10d')
'       100'
>>> format(total_user_score, '^10d')
'   100    '
>>> format(total_user_score, '<10d')
'100       '
>>> format(total_user_score, '>10d')
'       100'
```

Use nested with the print function:

```
>>> total_user_score = 100
>>> print(format(total_user_score, '10d'))
       100
>>> print(format(total_user_score, '^10d'))
   100
>>> print(format(total_user_score, '<10d'))
100
>>> print(format(total_user_score, '>10d'))
       100
```

Formatting Text (aligning the text and specifying a width)
Examples of use, nested with the print function (including output):

```
>>> format("You", '10s')
'You       '
>>> format("You", '^10s')
'   You    '
>>> format("You", '<10s')
'You       '
>>> format("You", '>10s')
'       You'

>>> print(format("You", '10s'))
You
>>> print(format("You", '^10s'))
   You
>>> print(format("You", '<10s'))
You
>>> print(format("You", '>10s'))
       You
```