ITE404:

Practical assignment: 20% of final grade

Due date: Wednesday 23rd of October

L**earning outcomes:**

7.1. Understand fundamental programming concepts and utilise relevant tools including: logic diagrams; text editor - command line or integrated development environment (IDE); compilers; interpreters.

7.2. Develop applications including simple data models, databases, data structures and programming, with attention to user experience and documentation.

Requirements: An electronic report submitted to Blackboard that includes documentation of the client's case study (in italics below), your solution, source code and any relevant screenshots. You source code must be in Python. Testing of your code will be done with wingIDE so while you are not required to upload the python files, you can include them in a zip file if you want to. You can write one program that addresses all the requirements, or you can write individual programs to demonstrate specific requirements, make sure your document and code are very clear on how your code works.

Case study:

Dax's Good Times Arcade (DGTA) has approached you asking you to create two games for the company and demonstrate that they have been tested properly. To do this you will write source code that addresses the requirements and produces the results per the testing specifications.

The games are:

Game 1:

"Two Dice": Two six-sided die are to be rolled and displayed, the total is displayed and a message about the total is displayed. If a 7 is rolled it should display "Exactly 7", otherwise "Better than 7" if the total is 8 to 12, and "Worse than 7" if the total is 2 to 6. You will need to ensure the value of each die is between 1 and 6 inclusive before it is processed.

"Test 1 for Two Dice"

The testing regime for this will be to firstly set the die values according to the table below and ensure it prints out the same values:

Die_1	Die_2	Total	Message displayed
1	4	5	Worst than 7
1	6	7	Exactly 7
6	1	7	Exactly 7
5	5	10	Better than 7

As a note, your code is not expected to print out the table itself, just each line in the table.

"Test 2 for Two Dice"

The next part of the testing regime will be to iterate through both dice, starting with die_1 = 1 and die_2 = 1. Your iteration should have the next test being die_1 = 1 and die_2 = 2, then die_1 =1 and die_2 = 2 and so on.

Each time it iterates and does a test, increase the test_count by 1.

Display each test taken, and at the end display the test_count as well. The test_count at the end is expected to be 36.

"Test 3 for Two Dice"

The final part of the test regime for this game is to conduct 36 000 tests where die_1 and die_2 are random numbers picked between 1 and 6 inclusively.

Use an array to store the results of this test: To do this create an array that has values 2 to 12 in it, and after each test, increment the appropriate array entry by 1. At the end of the 36 000 tests, display each array entry, and it's count, and then provide a total count at the end. As a note, you are only required to print the array entry and the count, your program does not need to draw or create a table.

As an example: Note this example is based on expected values, not an actual random sample: Your values will be slightly different for each number of occurrences but your total should still be 36 000.

Dice Total	Number of occurrences
2	1000
3	2000
4	3000
5	4000
6	5000
7	6000
8	5000
9	4000
10	3000
11	2000
12	1000

The total number of tests was 36000.

Game 2:

"33"

For this game, up to five players are dealt 3 cards from a standard deck of cards.

The cards 2 to 9 count as their face value, with Jacks, Queens, Kings counting as 10, and Aces counting as 13.

As DGTA is developing this game before taking it to market, your primary goal is to generate 3 cards for 1 player, and total them up according to the values above, display the cards on screen [in text format], and display the total and a message:

If the total is higher than 33 display "Too high, you lose.", if the total is exactly "33" display "Winner!" if the total is 28 to 32 display "A good total", if it is below 28 display "You still might win".

For our purposes, suits do not matter, but we still need them to ensure we can identify different cards.

Test 1 for "33"

To test this you will need to use the following values:

Card_1	Card_2	Card_3	Total	Message
Ten of Hearts	Ace of Hearts	Ace of Diamonds	36	Too high, you lose.
King of Clubs	Queens of Diamonds	Ace of Hearts	33	Winner!
Jack of Spades	Queens of Clubs	9 of clubs	29	A good total
8 of Hearts	6 of Spades	Queen of Spades	24	You still might win

Under normal testing regimes, you would then test and display all possible combinations of possible 3 card hands, starting with 2,2,2 and going to A,A,A this is not feasible for this assignment.

Instead DGTA wants you to run 5 tests where 3 cards are randomly selected from a deck, each card's name is displayed, the total of 3 cards is displayed, and the message is displayed.

Test 2 for "33"

For our purposes, it will probably be easier to use three different variables, and when we go to draw card_2 from the deck, we check to make sure it does not match card_1, and when we go to draw card_3 we check to make sure it does not match card_1 and card_2. If you are confident, you can use an array, however using 3 variables will be acceptable.

Test 3 for "33"

The final test is to do 10000 tests where 3 cards are randomly selected from the deck, their total is calculated and the array value is incremented by 1.

As an example, if the 2 of hearts, 2 of clubs and 2 of diamonds were selected then array[6] would be incremented. If the 2 of hearts, the ten of clubs and the King of spades were selected then array[22]

would be incremented. If the Ace of Hearts, Ace of Spades, Ace of diamonds were selected then array[39] would be incremented.

The only output requirement for this test is:

At the end of the tests display the total number of times there was an exact winner [where the total is 33].

Report requirements

Your report has the following requirements:

Page numbers and Table of contents (This will be software dependent)

Introduction stating that this is an assignment for ITE404, a description of what the assignment is, and a description of order of material. Any code or requirement that does not work should be stated in the introduction. Your client also wants a brief explanation of how your code selects random numbers (for this you only need to describe which random number selector you used and why you used it).

A section discussing the "Two dice" game, and showing the output required. The output can either be a screenshot, or a copy or paste from running your code.

A section discussing the "33" game, and showing the output required. The output can either be a screenshot, or a copy or paste from running your code.

For this section you will need to explain how your code selects three cards that are different each time you do a test.

You will need to explain how your code resets the deck [or ensures all cards could be picked from the deck] for each test.

Special report requirement:

DGTA has a special requirement that you explain which random number selection function you used and why, how your code selects a card, how you generate the name of a specific card, and how you reset test 3 in "33". These requirements will need to be addressed in your report.

Other notes:

All code will require good commenting: For this assignment this means each function will need a brief description of any inputs the function takes, any outputs the function returns, and what the overall purpose of the function is.

For arrays, the array will need a description for what values it holds, and why it holds them.

For any loops, comments explaining why the loop is being used will be needed.

The marking of the "code works" will be based on the following schedule:

Non-existent or does not work at all [longer than 3 minutes to fix]	0
Exists and needs some minor changes to work [less than 3 minutes to fix]	0.5
Exists and works as intended	1

Mark Schedule

Mark Schedule:	**Possible:**	**Given**	
Report requirements are met	1		
Report has a good clear introduction	1		
Game 1 description of solution	1		
Game 2 description of solution	1		
Game 1: Two dice			
Test 1			
Code comments	0.5		
Code works	1		
Output	0.5		
Test 2			
Code comments	0.5		
Code works	1		
Output	0.5		
Test 3			
Code comments	0.5		
Code works	1		
Output	0.5		
Game 2: 33			
Test 1			
Code comments	0.5		
Code works	1		
Output	0.5		
Test 2:			
Code comments	0.5		
Code works	1		
output	1		
Test 3			
Code comments	0.5		
Code works	1		
Output	1		
Explanation of how random numbers are selected for this assignment	1		
Specifics of how a card is selected	1		
Explanation of how “Reset of card selection between tests” works	0.5		
Explanation of how the name of a card is created	0.5		