CompSci 101 Lab 3 — Functions

Topics covered:

- Defining functions
- Breaking a program up into functions with each function handling one task only
- Calling functions

Programming Exercises **(Out of 10 marks - Submit functions on CodeRunner3)**

IMPORTANT: for each of your programs you need to add a docstring at the top of the program. The docstring should contain your name, your username, date and a short description of the program.

Questions 1 - 5. Program 1

The skeleton code for the `Lab03Program1.py` program is provided. In the program there are five functions which need to be completed by you. As you complete each function you can test that the function is correct by pasting the whole function (including the header) into CodeRunner3 and pressing the CHECK button.

1. `get_number_from_user(prompt)`
This function is passed a string as a parameter. The function obtains an integer from the user using the parameter string as the prompt. The function returns the integer entered by the user.

2. `get_ticket_price(number_of_tickets, ticket_price)`
This function is passed two parameters: an integer, the number of tickets (`number_of_tickets`), and a float, the price of each ticket (`ticket_price`). The function returns the total cost of the tickets rounded to the nearest whole number.

3. `get_discount(number_of_tickets, total_price)`
This function is passed two parameters: an integer, the number of tickets (`number_of_tickets`), and a float, the price of all the tickets (`total_price`). The function returns the discount amount rounded to the nearest whole number. The discount is the **bigger** of the following two calculations:

- either $4 discount for each ticket,
- or 10% of the total price.

4. `get_gst_amount(price)`
This function is passed a float as a parameter. The function returns the GST amount rounded to two decimal places. The GST amount is 15% of the `price` parameter.

5. `display_ticket_price(tickets, price, discount)`
This function is passed three integer parameters: the number of tickets (`tickets`), the full price of all the tickets (price), and the discount amount (`discount`). The function calculates:

- the amount to be paid after the discount has been subtracted from the discount,
- the GST portion of the amount to be paid. Your code **MUST** make a call to the `get_gst_amount()` function developed in Part d).

and then the function prints 5 lines of output:

- Lines 1 and 5 are 20 `"*"` symbols,
- Line 2 prints the string `"Tickets: "` followed by the number of tickets,
- Line 3 prints the string `"Price: $"` followed by the price of the tickets. Line 3 ends with the string `" (discount included: $"` followed by the discount amount followed by `").",`
- Line 4 prints the string `"GST included: $"` followed by the GST amount.

Example output using the completed program is shown below. The output of your program should have the same format (the user input is shown in bold in a larger font):

```
********************
Tickets: 7
Price: $77 (discount included: $28).
GST included: $11.55
********************
```

Question 6 Program 2

The skeleton code for the `Lab03Program2.py` program is provided. Define the `get_first_last_number(number)` function which is passed an integer as a parameter and returns the two digit integer made up of the first digit followed by the last digit of the parameter number.

Notes

If the `number` parameter is a negative number, the function takes the absolute value of the parameter number and then proceeds as for a positive number,
If the `number` parameter is a single digit number, the function returns the `number` parameter repeated.
If the `number` parameter is zero, the function returns zero.

For example:

```
print(get_first_last_number(24678))
print(get_first_last_number(6))
print(get_first_last_number(-19876))
```

prints

```
28
66
16
```

Question 7 Program 3

Some parts of the skeleton code for the `Lab03Program3.py` program are shown below. Three functions in the program have been written for you. Do not change these functions in any way.

You need to complete the `main()` function which calls each of the existing functions to achieve the tasks listed below (each task is a call to one of the three functions).

- Obtain a number from the user (in the call to the function, use the `request` variable defined in the `main()` function),
- Calculate the cost of the items,
- Print the cost details.

```
def main():
    request = "Enter number (5 - 20): "
    handling_cost = 5
    cost_per_item = 4.25

def get_number_of_items(prompt):
    #prompts the user and returns the integer entered by the user

def get_cost(number_of_items, cost_per_unit, handling_cost):
    #returns the total cost of the items including packaging costs

def display_details(items, cost_each, handling_cost, final_price):
    #displays three lines of information about the total cost of the items

main()
```

The expected output of the completed program, if the user enters 10 at the prompt, is shown below (the user input, 10, is shown in a larger font in bold):

```
Enter number (5 – 20): 10
Items: 10 Cost per item: $4.25
Handling cost: $5
Total: $48
```