



ICTPRG302

Apply introductory programming techniques

Assessment 2 of 6

Case Study & Demonstration

Assessor Guide



Assessment Instructions

Task Overview

This assessment consists of one [1] task.

Important: Before commencing your work, you must update your *Student name* and *Student number* in the footer from **page 2** onwards.

Additional Resources and Supporting Documents

To complete this assessment, you will need:

- Learning Material
- Visual Studio Code

Assessment Information

Submission

You are entitled to three [3] attempts to complete this assessment satisfactorily. Incomplete assessments will not be marked and will count as one of your three attempts.

All questions must be responded to correctly to be assessed as satisfactory for this assessment.

Answers must be typed into the space provided and submitted electronically via the Learning Platform. Hand-written assessments will not be accepted unless previously arranged with your assessor.

Reasonable adjustment

Students may request a reasonable adjustment for assessment tasks.

Reasonable adjustment usually involves varying:

- the processes for conducting the assessment (e.g. allowing additional time)
- the evidence gathering techniques (e.g. oral rather than written questioning, use of a scribe, modifications to equipment)

However, the evidence collected must allow the student to demonstrate all requirements of the unit.

Refer to the Student Handbook or contact your Trainer for further information.



Please consider the environment before printing this assessment.

Case Study

You are an ICT technician at Bounce Fitness, a premier provider of fitness and wellness in Australia. Bounce Fitness prides itself in an accurate and efficient system to keep track of member records.

One of your responsibilities is to develop a simple program to track the attendance of members for various fitness classes. You have already written the initial code for the attendance tracker, but it contains errors as the process did not complete successfully upon testing.

You must now:

- Examine the variables of the code and conduct debugging techniques to identify the errors
- Make the corrections necessary to address the identified errors

Below is a snippet of the Python code for the attendance tracker:

```
# Welcome to Bounce Fitness Attendance Tracker!
# Create an empty dictionary to store class attendance
attendance = {}
# Function to mark attendance for a class
def mark_attendance(class_name, member_name):
    if class_name in attendance:
        attendance[class_name].append(member_name)
    else:
        attendance[class_name] = [member_name]
# Main program
while True:
    print("Select an option:")
    print("1. Mark Attendance")
    print("2. View Attendance")
    print("3. Exit")
    choice = input("Enter your choice: ")
    if choice == '1':
        class_name = input("Enter the class name: ")
        member_name = input("Enter the member's name: ")
        mark_attendance(class_name, member_name)
        print(f"{member_name} is marked as present in {class_name} class.")
    elif choice == '2':
        print("Classwise Attendance:")
        for class_name, members in attendance.items():
            print(f"{class_name}: {' '.join(members)}")
```

```
elif choice == '3':  
    print("Exiting Attendance Tracker. Goodbye!")  
    break  
else:  
    print("Invalid choice. Please select a valid option.")
```

Task 1

Note: For this task, you will need to use Visual Studio Code. Please access the software here:

<https://code.visualstudio.com/download>

Record your screen while examining the variable contents and conducting debugging techniques to the provided Python code.

STEPS TO TAKE

1. Start the recording by stating your name
2. Open the code in a Visual Studio Code
3. Examine the variable content of the code using features of the programming software.
4. Debug the code to identify all errors using features of the Visual Studio Code.
5. Outline all identified errors in Table 1 below.
6. Edit the code to correct all identified errors.
7. Confirm that the code is functioning as intended following the edits made.
8. Save and submit your video using the following naming convention:

<Student Name>_Video 1_Debugging

During this process, you will need to ensure you perform all the steps listed in the **Assessor Observation Form** below.

Assessor instructions: Students must examine the variable contents and conduct debugging techniques to the provided Python code. They must also outline all the identified errors in Table 1 below.

The acceptable responses must:

- Address all the requirements mentioned in the Assessor Observation Checklist.
- Reflect the characteristics described in the benchmark answer below in Table 1.

The benchmark answer is provided below.

Table 1

Outline all the errors found in the provided code.

Assessor instructions: Students must outline all the errors found in the provided code.

Although wording will vary, for satisfactory performance, responses must correspond to the issues or mistakes with the Python code for Bounce Fitness' attendance tracker that will hinder it from working properly.

a.

The code inside the if class_name in attendance block appends member_name to the list of attendees, but it should append it to the dictionary's value (the list of attendees for that class).

b.

There is no mechanism for handling invalid choices (choices other than '1', '2', and '3'). The program simply prints an error message and continues.

ASSESSOR OBSERVATION FORM

During this case study task:	YES/NO	Date Observed	Assessor's comments
1. The student opens the code in a programming software.	<input type="checkbox"/> YES <input type="checkbox"/> NO		
2. The student uses the features of the programming software in examining the variable content of the provided Python code. In demonstrating this:			
a. Student sets breakpoints in the sections where variable content is going to be checked. Breakpoints refer to markers that pause the application execution when it reaches a certain line.	<input type="checkbox"/> YES <input type="checkbox"/> NO		
b. Student starts the debugging mode of the programming software. For example, this is started by clicking the 'Debug' button.	<input type="checkbox"/> YES <input type="checkbox"/> NO		
c. Student checks the variable values once the application pauses as it reaches a breakpoint. For example, the student looks at the panel that displays the values of the variable at the current point.	<input type="checkbox"/> YES <input type="checkbox"/> NO		

d. Student changes the code based on the detected errors in the variable content of the code.	<input type="checkbox"/> YES <input type="checkbox"/> NO
3. The student opens the Python code in an integrated development environment (IDE).	<input type="checkbox"/> YES <input type="checkbox"/> NO
4. The student uses the IDE to start debugging the code. <i>Assessor to tick N/A if the student does not use the specific debugging technique.</i> In demonstrating this, the student must perform AT LEAST TWO of the following debugging techniques:	
a. Student uses the debugging tools available in the IDE to debug the code.	<input type="checkbox"/> YES <input type="checkbox"/> NO <input type="checkbox"/> N/A
b. Student goes through the code line by line to perform interactive debugging.	<input type="checkbox"/> YES <input type="checkbox"/> NO <input type="checkbox"/> N/A
c. Student puts conditional statements in particular sections of the code to be checked to perform conditional debugging.	<input type="checkbox"/> YES <input type="checkbox"/> NO <input type="checkbox"/> N/A
5. The student edits the Python code to correct the errors detected through the debugging techniques.	<input type="checkbox"/> YES <input type="checkbox"/> NO
6. The student runs the application again to confirm that the changes made addressed the errors in the code.	<input type="checkbox"/> YES <input type="checkbox"/> NO
7. The student continues to examine variable content, perform debugging techniques and run the application until the application is functioning as it is intended to. <i>Assessor to tick N/A if the code was corrected during the first attempt to change the code.</i>	<input type="checkbox"/> YES <input type="checkbox"/> NO <input type="checkbox"/> N/A

Assessment submission checklist

Students must have completed all questions within this assessment before submitting. This includes:

1	Task 1 – Video Recording	<input type="checkbox"/>
---	--------------------------	--------------------------

Assessment feedback

Assessors are to indicate the assessment outcome as Satisfactory (S) or Not Yet Satisfactory (NYS).

Assessor comments:	<input type="checkbox"/> S	<input type="checkbox"/> NYS
---------------------------	----------------------------	------------------------------


Congratulations, you have reached the end of Assessment 2!

© 2023 Precision Group (Australia)

No part of this resource may be reproduced in any form or by any means, electronic or mechanical, including photocopying or recording, or by an information retrieval system without written permission from Precision Group

(Australia). Legal action may be taken against any person who infringes their copyright through unauthorised copying.

These terms are subject to the conditions prescribed under the Australian Copyright Act 1968.

© UP Education Online Pty Ltd 2023

Except as permitted by the copyright law applicable to you, you may not reproduce or communicate any of the content on this website, including files downloadable from this website, without the permission of the copyright owner.

WARNING

This material has been reproduced and communicated to you by or on behalf of UP Education in accordance with section 113P of the *Copyright Act* 1968 [the Act].

The material in this communication may be subject to copyright under the Act. Any further reproduction or communication of this material by you may be the subject of copyright protection under the Act.

Do not remove this notice.