

Table of contents

Extended Learning Portfolio	1
ISYS2001 Introduction to Business Programming	1
School of Management	1
Semester 1 2024	1
Examination Instructions	1
Question 1: Monthly Budget Planner	2
Question 2: Code City Adventures Refactoring	3
Question 3: Code City Adventures Implementation	4
Question 4: Reflective Report	5

Extended Learning Portfolio

ISYS2001 Introduction to Business Programming

School of Management

Semester 1 2024

Examination Instructions

Format: This examination is open-book. You may use textbooks, web content, and AI tools to complete the exam. All submitted work must be your own, and any external ideas or work must be properly referenced.

Collaboration: Do not discuss your responses with fellow students. For any inquiries about the questions, contact the instructor directly. All questions and responses will be posted on the discussion forum.

Duration: You have 24 hours to complete the exam, starting from the official exam start time. This timeframe is fixed, regardless of when you begin. If you have a CAP arrangement, your exam duration will be adjusted accordingly. Contact me immediately if you believe your CAP accommodations are not adequately addressed.

Questions: The exam consists of four questions. Answer all of them. Each question should be in its own notebook, except for Question 4, which can be in a Microsoft Word document or PDF.

Use of AI Tools: Transparent and ethical use of Generative AI (Gen-AI) is allowed and expected. Include either copies of all AI conversations or a document containing links to the conversations (if your AI tool provides this). For each question, indicate or reference the relevant AI conversation.

Python Code: All Python code must be executable on the Google Colab platform. Ensure that any additional data, files, or code required by your notebook are downloaded and imported within the notebook itself.

Submission Instructions:

1. Create a private GitHub repository.
2. Upload all your responses, including the Word document or PDF for Question 4, to this repository.
3. Download a zip file of your GitHub repository.
4. Submit this zip file via the link on Blackboard.
5. Submit Question 4 separately via the Turnitin link on Blackboard as either a Word document or a PDF.

Marking Scheme: The exam is worth 100 marks in total, with each question allocated the following marks:

- Question 1: 25 Marks
- Question 2: 20 Marks
- Question 3: 40 Marks
- Question 4: 15 Marks

Guidelines: Ensure your responses are well-structured, clearly written, and demonstrate your understanding of the course material.

Good luck!

Question 1: Monthly Budget Planner

Instructions:

Design and implement an application that helps users create and manage a monthly budget. The application should prompt the user to input their monthly income and various expense categories (e.g., rent, groceries, utilities, entertainment). Your design should:

1. Prompt the user for their monthly income and amounts for each expense category.
2. Calculate and display the total expenses and the remaining balance after all expenses.
3. Allow the user to repeat the process with different inputs to plan different budget scenarios.

Your design should demonstrate the first five steps of the development methodology used in ISYS2001. You can write your design in a Word document or a notebook. Create a notebook called `monthly_budget.ipynb` to implement your design. Your implementation should follow best practices and demonstrate the last two steps of the methodology. This includes at least input validation and testing.

[25 Marks]

Question 2: Code City Adventures Refactoring

Background:

“Code City Adventures” is a text-based adventure game where players use Python to solve challenges in a modern city setting. In the first level, “The Neighbourhood Watch,” the player helps organise a schedule for volunteer watch shifts.

Instructions:

You’ve been tasked with refactoring a part of the “Code City Adventures” game (Level 1: The Neighbourhood Watch). The current code snippet (provided below) is functional but could be improved. Your goal is to refactor the code to meet industry standards while maintaining the game’s core functionality.

Code Snippet:

```
def schedule_volunteers(volunteers: list, shifts: list) -> dict:
    """
    Assigns volunteers to available shifts based on their preferences.

    >>> schedule_volunteers(["Alice", "Bob"], ["Morning", "Afternoon"])
    {'Morning': 'Alice', 'Afternoon': 'Bob'}
    """

    # Initialize an empty dictionary for assignments
    assignments = {}

    for v in volunteers: # Loop through the volunteers
        # Assuming user has entered a valid shift number
        # Get the shift preference
        preference_num = int(input(f"{v}, enter your preferred shift number (1-{len(shifts)})"))
        preference_num -= 1 # Convert to 0-based index
```

```

        assignments.append(shifts.pop(preference_num), v)
    return assignments

if __name__ == "__main__":
    # List of volunteers
    vols = ["Alice", "Bob"]
    # List of shifts
    shifts = ["Morning", "Afternoon"]

    # Call the schedule_volunteers function
    result = schedule_volunteers(vols, shifts) # Typo in variable name

    # Print the results
    print(result)

```

Tasks:

1. Analyse the provided code and identify areas where it deviates from best practices.
2. Refactor the code, applying appropriate improvements like:
 - Meaningful variable names
 - Improved function organisation
 - Clearer comments
 - Error handling (if applicable)
 - Any additional enhancements you deem necessary
3. Provide a brief explanation of the changes you made and why they improve the code.

[20 Marks]

Question 3: Code City Adventures Implementation

Background:

“Code City Adventures” is a text-based adventure game where players use Python to solve challenges in a modern city setting. In the first level, “The Neighbourhood Watch,” the player helps organise a schedule for volunteer watch shifts. In the second level “The Automated Cafe”, the player helps a café automate its order system using conditionals to handle menu choices and calculate costs.

Instructions:

Implement the first level (The Neighbourhood Watch) and the second level (The Automated Cafe) of the “Code City Adventures” game in one or more Google Colab notebooks. Ensure your implementation includes:

- Clear instructions for the player
- Text-based input and output
- Use of variables, input, loops, and functions
- Proper testing with doctests and/or assertions
- Clear documentation (comments and docstrings)

[40 Marks]

Question 4: Reflective Report

Instructions:

Write a reflective report that identifies and discusses what you perceive as the most impactful activity within this course unit, and its contributions to your understanding of an ISYS2001 activity or topic. Additionally, please incorporate all your weekly journal entries as an appendix to this report. The report should be included in your GitHub repository and submitted either as a Microsoft Word document or PDF via the Turnitin link available on Blackboard.

[15 Marks]
