ST 1/ST1 G Major Assignment
Programming Projects

Assignment submission due date: Week 12 Friday 11.59 pm
This assignment is worth 30 marks

Group of 1 (Solo): Part A and Part B
Group of 2 Students: Part A, Part B and Part C/D/E
Group of 3 Students: Part A, Part B, and any one of Part C/D/E
Group of 4 Students: Part A, Part B, and any two of Part C/D/E

Submission Requirements for all projects
Each Project needs to include following project deliverables:

1. Python code files
2. Assignment Report (word or pdf) with following items:
    a. Requirements Analysis:(Describe the problem briefly in your own words (~100 words), for which you will develop a python software tool/solution)
    b. Algorithm Design (either pseudo code or flowchart)
    c. Hierarchy Chart / UML (UML if you have classes, and hierarchy chart if you have functions)
    d. Brief Code Walkthrough – Console & GUI Program(Explanation of code fragments along with code listing)
    e. Evidence of Testing (Test plan and associated screenshots)
    f. Special Features (Describe any special features used in your code such as input validation, exception handling, more than the specified learning objective of Python constructs etc.)
    g. Reflection (Brief reflection on lessons learnt by completing this activity (~100 words))

Submit the Python code files and Major Assignment report in Canvas Major Assignment submission link before the due date (Week 12 Friday 11.59 pm).

Part A
Project 1: Payroll Calculator

- **Learning Objective: Using Basic Python Input Output and Arithmetic Statements**

Problem Description:

Write a Python program (Console and GUI) that reads the following information from user, and prints a payroll statement:

Employee's name (e.g., Smith)
Number of hours worked in a week (e.g., 10)
Hourly pay rate (e.g., 60.75)
ATO tax withholding rate (e.g., 30%)
Medicare Levy rate (e.g., 2%)

Here is some sample acceptance test data for console program:

<Output>

```
Enter employee's name: John Smith
```

```
Enter number of hours worked in a week: 10
Enter hourly pay rate: 60.75
Enter ATO tax withholding rate: 0.30
Enter Medicare Levy rate: 0.02
```

Employee Name: Smith
Hours Worked: 10.0
Pay Rate: $60.75
Gross Pay: $607.50
Deductions:
ATO tax (30.0%): $182.25
Medicare Levy (2.0%): $12.15
Total Deduction12: $194.4

Net Pay: $413.10
<End Output>

Project 2: Restaurant Selector

- **Learning Objective: Using Python Conditional and Branching Statements**

Problem Description:

You have a group of friends coming to visit for your high school reunion, and you want to take them out to eat at a local restaurant. You are not sure if any of them have dietary restrictions, but your restaurant choices are as follows:

- **Joe's Gourmet Burgers—Vegetarian: No, Vegan: No, Gluten-Free: No**
- **Main Street Pizza Company—Vegetarian: Yes, Vegan: No, Gluten-Free: Yes**
- **Corner Café—Vegetarian: Yes, Vegan: Yes, Gluten-Free: Yes**
- **Mama's Fine Italian—Vegetarian: Yes, Vegan: No, Gluten-Free: No**
- **The Chef's Kitchen—Vegetarian: Yes, Vegan: Yes, Gluten-Free: Yes**

Write a Python program (Console and GUI) that asks whether any members of your party are vegetarian, vegan, or gluten-free, to which then displays only the restaurants to which you may take the group.

Here is an example of the program's output:

```
Is anyone in your party a vegetarian? yes Enter
Is anyone in your party a vegan? no Enter
Is anyone in your party gluten-free? yes Enter
Here are your restaurant choices:
   Main Street Pizza Company
   Corner Cafe
   The Chef's Kitchen
```

Here is another example of the program's output:

```
Is anyone in your party a vegetarian? yes Enter
Is anyone in your party a vegan? yes Enter
Is anyone in your party gluten-free? yes Enter
Here are your restaurant choices:
   Corner Cafe
   The Chef's Kitchen
```

Project 3: Population Tracker

- **Learning Objective: Using Python Repetition and Loop Statements**

Problem Description

Write a Python program (Console and GUI) that predicts the approximate size of a population of organisms.

The application should use text boxes to allow the user to enter the following information:

- starting number of organisms,
- the average daily population increase (as a percentage), and
- the number of days the organisms will be left to multiply.

For example, assume the user enters the following values:

```
Starting number of organisms: 2
Average daily increase: 30%
Number of days to multiply: 10
The program should display the following table of data
```

Day Approximate	Population
1	2
2	2.6
3	3.38
4	4.394
5	5.7122
6	7.42586
7	9.653619
8	12.5497
9	16.31462
10	21.209

Project 4: Password Checker

- **Learning Objective: Using Python Methods and Functions**

Problem Description:

Some Websites impose certain rules for passwords, involving use of certain numbers and special characters.

Write a Python program (Console and GUI) that prompts the user to enter a password and displays "*valid password*" if the rule is follows rules of the company, or "*invalid password*" otherwise.

Write a method or a function for implementing the password checking rules for the method/function as follows:

- A password must have at least eight characters.
- A password must consist of only letters and digits.
- A password must contain at least two digits.

Sample 1

```
Enter a string for password: wewew43x
valid password
```

Sample 2

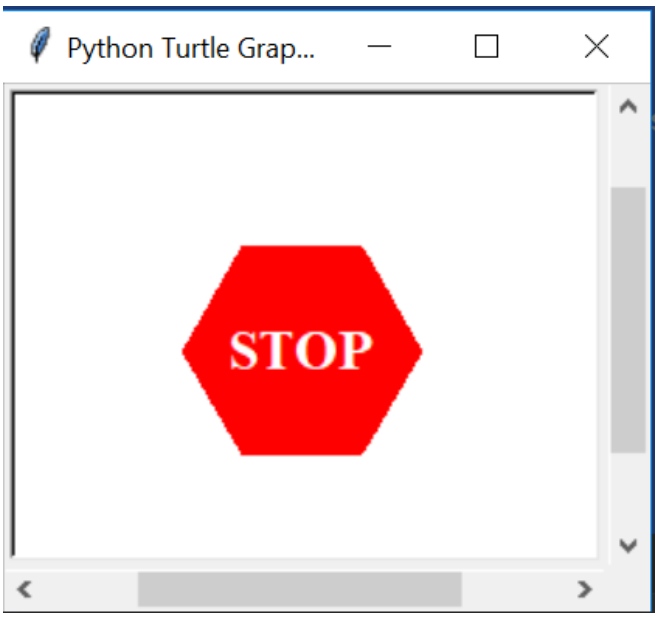
```
Enter a string for password: 343a
invalid password
```

Project 5: Turtle Graphics

- **Objective: Using Turtle Graphics Class and its Methods**

Problem Description

Write a Python program using Turtle graphics library, that displays a STOP sign, as shown in Figure below. The hexagon needs to be in red and the text needs to be in white colour.



Part B

Project 1: Video Time Logger

- **Learning Objective: Working with Files in**

PythonProblem Description:

Kevin is a freelance video producer who makes TV commercials for local businesses. When he makes a commercial, he usually films several short videos. Later, he puts these short videos together to make the final commercial. He has asked you to write the following two programs.

1. A program that allows him to enter the running time (in seconds) of each short video in a project. The running times are saved to a file.

2. A program that reads the contents of the file, displays the running times, and then displays the total running time of all the segments.

Project 2: Fitness Tracker

- **Learning Objective: Using Python Lists, Collections and**

FilesProblem Description:

A Personal Fitness Tracker is a wearable device that tracks your physical activity, calories burned, heart rate, sleeping patterns, and so on. One common physical activity that most of these devices track is the number of steps you take each day.

- The file "*steps.txt*" at the following web link contains the number of steps a person has taken each day for a year.

- There are 365 lines in the file, and each line contains the number of steps taken during a day. (The first line is the number of steps taken on January 1st, the second line is the number of steps taken on January 2nd, and so forth.)

- Write a program that reads the file, then displays the average number of steps taken for each month.

- (The data is from a year that was not a leap year, so February has 28 days.)

  https://drive.google.com/file/d/1fBVZlrVC69MMkN-y1ICdUk2JVt4ia4L0/view?usp=sharing

Project 3: Employee Management System

- **Learning Objective: Using Python Classes and**

Objects, Inheritance and Polymorphism

Problem Description:

In this problem, you will develop an employee management system using Python. Here are the brief requirements of the program:

1. Write a class named `Employee`, that holds information about each employee in attributes such as name, ID number, department, and job title.

2. Once you have written the class, Draw the UML diagram, and write a program that creates three objects of employee class, with following information and displays their details.

Name	ID Number	Department	Job Title
Susanna Myer	47899	Accounting	Vice President
Mark Joseph	39119	Info Tech	Programmer
Joyce Roberts	81774	Manufacturing	Engineer

3. Create a new class named **ShiftEmployee,** that is a subclass of Employee class. The shift employee has additional attributes in addition to name, ID number, Department and Job Title. The additional attributes are:

    - Shift number (an integer, such as 1,2, or 3)
    - Hourly pay rate

The workday is divided into two shifts: day and night. The shift attribute will hold an integer value representing the shift that the employee works. The day shift is shift 1 and night shift is shift 2.

4. Once you have written the classes, write a program that creates an object of ShiftEmployee class and prompts the user to enter data for each of the object's data attributes. Store the data in the object, then use the object's accessor and mutator methods to retrieve it and display it on the screen. Demonstrate the ShiftEmployee class by creating atleast 2 shift employee objects.

5. Create another new class named **Contractor,** that is a subclass of Employee class. The contractor has additional attributes in addition to name, ID number, Department and Job Title. The additional attributes are:

    a. Contract End Date
    b. Australian Business Number (ABN)
    c. Fixed Contract Salary until the end of contract

Demonstrate this class by creating 2 contractor objects.

Part C:

- **Learning Objective: Using Python Packages and Modules (PyGame Library)**

Alien Invasion Game Project

Problem Description:

This is just a brief description. Detailed requirements will be discussed with individual teams/groups working on this assignment task.

- In Alien Invasion, the player needs to control a rocket ship that appears at the bottom center of the screen.
- The player can move the ship right and left using the arrow keys and shoot bullets using the spacebar.
- When the game begins, a fleet of aliens fills the sky and moves across and down the screen.
- The player shoots and destroys the aliens. If the player shoots all the aliens, a new fleet appears that moves faster than the previous fleet.
- If any alien hits the player's ship or reaches the bottom of the screen, the player loses a ship.
- If the player loses three ships, the game ends.

Part D:

- **Learning Objective: Using Python Packages and Modules (Matplotlib and PlotlyLibrary)**

Data Analysis and Visualisation Project

Problem Description:

This is just a brief description. Detailed requirements will be discussed with individualteams/groups working on this assignment task.

- In this problem you will work on two different tasks, involving downloading data setsfrom online sources and create working visualizations of that data.
- The ability to analyze this data allows you to discover patterns and connections thatno one else has found.
- You will access and visualize data to process **weather data** stored in the CSV (comma-separated values) format and analyze high and low temperatures over time intwo different locations in the world.
- You will need to use Matplotlib to generate a chart based on our downloaded data todisplay variations in temperature in two dissimilar environments.
- The second task involves accessing **earthquake data** and use Plotly to draw a worldmap showing the locations and magnitudes of recent earthquakes.

Part E:

- **Learning Objective: Using Python Packages and Modules (Django for WebApplication)**

Django Web Application Project

Problem Description:

This is just a brief description. Detailed requirements will be discussed with individualteams/groups working on this assignment task.

- In this problem, you will develop an online Web application called **Learning Journal**that allows users to log the topics they're interested in and to make journal entries as they learn about each topic.
- The Learning Log home page will describe the site and invite users to either registeror log in.
- Once logged in, a user can create new topics, add new entries, and read and editexisting entries.

ST1(4483) / ST1G(8995) Major Assignment Marking Rubric

Major Assignment submission due date: Week 12 Friday 11.59 pm

This assignment is worth 30 marks

Student Name: **Student Name:** **Student Name:** **Student Name:**	**Unit: 4483/8995**	**Student ID** **Student ID** **Student ID** **Student ID**		
Assignment Project ID	Assessment Criteria	**Maximum Marks**	**Marks Obtained**	**Comments**
Part A **Project 1** **(3 marks)**	**1. Requirements Analysis:** (Describe the problem briefly in your own words (~100 words), for which you will develop a python software tool/solution)			
	2. Algorithm Design (either pseudo code or flowchart)			
	3. Hierarchy Chart / UML (UML if you have classes, and hierarchy chart if you have functions)			
	4. Brief Code Walkthrough – Console & GUI Program (Explanation of code fragments along with code listing)			
	5. Evidence of Testing (Test plan and associated screenshots**)**			
	6. Special Features (Describe any special features used in your code such as input validation, exception handling etc.)			
	7. Reflection Brief reflection on lessons learnt by completing this activity (~100 words)			

Part A **Project 2** **(3 marks)**	**1. Requirements Analysis:** (Describe the problem briefly in your own words (~100 words), for which you will develop a python software tool/solution)			
	2. Algorithm Design (either pseudo code or flowchart)			
	3. Hierarchy Chart / UML (UML if you have classes, and hierarchy chart if you have functions)			
	4. Brief Code Walkthrough – Console & GUI Program (Explanation of code fragments along with code listing)			
	5. Evidence of Testing (Test plan and associated screenshots**)**			
	6. Special Features (Describe any special features used in your code such as input validation, exception handling etc.)			
	7. Reflection Brief reflection on lessons learnt by completing this activity (~100 words)			

Part A **Project 3** **(4 marks)**	**1. Requirements Analysis:** (Describe the problem briefly in your own words (~100 words), for which you will develop a python software tool/solution)			
	2. Algorithm Design (either pseudo code or flowchart)			
	3. Hierarchy Chart / UML (UML if you have classes, and hierarchy chart if you have functions)			

	4. Brief Code Walkthrough – Console & GUI Program (Explanation of code fragments along with code listing)			
	5. Evidence of Testing (Test plan and associated screenshots**)**			
	6. Special Features (Describe any special features used in your code such as input validation, exception handling etc.)			
	7. Reflection Brief reflection on lessons learnt by completing this activity (~100 words)			
Part A **Project 4** **(4 marks)**	**1. Requirements Analysis:** (Describe the problem briefly in your own words (~100 words), for which you will develop a python software tool/solution)			
	2. Algorithm Design (either pseudo code or flowchart)			
	3. Hierarchy Chart / UML (UML if you have classes, and hierarchy chart if you have functions)			
	4. Brief Code Walkthrough – Console & GUI Program (Explanation of code fragments along with code listing)			
	5. Evidence of Testing (Test plan and associated screenshots)			
	6. Special Features (Describe any special features used in your code such as input validation, exception handling etc.)			
	7. Reflection			

	Brief reflection on lessons learnt by completing this activity (~100 words)			

Part A **Project 5** **(4 marks)**	**1. Requirements Analysis:** (Describe the problem briefly in your own words (~100 words), for which you will develop a python software tool/solution)			
	2. Algorithm Design (either pseudo code or flowchart)			
	3. Hierarchy Chart / UML (UML if you have classes, and hierarchy chart if you have functions)			
	4. Brief Code Walkthrough – Console & GUI Program (Explanation of code fragments along with code listing)			
	5. Evidence of Testing (Test plan and associated screenshots**)**			
	6. Special Features (Describe any special features used in your code such as input validation, exception handling etc.)			
	7. Reflection Brief reflection on lessons learnt by completing this activity (~100 words)			

Part B **Project 1** **(4 marks)**	**1. Requirements Analysis:** (Describe the problem briefly in your own words (~100 words), for which you will develop a python software tool/solution)			
	2. Algorithm Design (either pseudo code or flowchart)			

	3. Hierarchy Chart / UML (UML if you have classes, and hierarchy chart if you have functions)			
	4. Brief Code Walkthrough – Console & GUI Program (Explanation of code fragments along with code listing)			
	5. Evidence of Testing (Test plan and associated screenshots**)**			
	6. Special Features (Describe any special features used in your code such as input validation, exception handling etc.)			
	7. Reflection Brief reflection on lessons learnt by completing this activity (~100 words)			

Part B **Project 2** **(4 marks)**	**1. Requirements Analysis:** (Describe the problem briefly in your own words (~100 words), for which you will develop a python software tool/solution)			
	2. Algorithm Design (either pseudo code or flowchart)			
	3. Hierarchy Chart / UML (UML if you have classes, and hierarchy chart if you have functions)			
	4. Brief Code Walkthrough – Console & GUI Program (Explanation of code fragments along with code listing)			
	5. Evidence of Testing (Test plan and associated screenshots**)**			

	6. Special Features (Describe any special features used in your code such as input validation, exception handling etc.)			
	7. Reflection Brief reflection on lessons learnt by completing this activity (~100 words)			

Part B **Project 3** **(4 marks)**	**1. Requirements Analysis:** (Describe the problem briefly in your own words (~100 words), for which you will develop a python software tool/solution)			
	2. Algorithm Design (either pseudo code or flowchart)			
	3. Hierarchy Chart / UML (UML if you have classes, and hierarchy chart if you have functions)			
	4. Brief Code Walkthrough – Console & GUI Program (Explanation of code fragments along with code listing)			
	5. Evidence of Testing (Test plan and associated screenshots**)**			
	6. Special Features (Describe any special features used in your code such as input validation, exception handling etc.)			
	7. Reflection Brief reflection on lessons learnt by completing this activity (~100 words)			

Part C **(5 marks)**	**1. Requirements Analysis:** (Describe the problem briefly in your own words (~100 words), for which you will develop a python software tool/solution)			
	2. Algorithm Design (either pseudo code or flowchart)			

	3. Hierarchy Chart / UML (UML if you have classes, and hierarchy chart if you have functions)			
	4. Brief Code Walkthrough – Console & GUI Program (Explanation of code fragments along with code listing)			
	5. Evidence of Testing (Test plan and associated screenshots**)**			
	6. Special Features (Describe any special features used in your code such as input validation, exception handling etc.)			
	7. Reflection Brief reflection on lessons learnt by completing this activity (~100 words)			

Part D **(5 marks)**	**1. Requirements Analysis:** (Describe the problem briefly in your own words (~100 words), for which you will develop a python software tool/solution)			
	2. Algorithm Design (either pseudo code or flowchart)			
	3. Hierarchy Chart / UML (UML if you have classes, and hierarchy chart if you have functions)			
	4. Brief Code Walkthrough – Console & GUI Program (Explanation of code fragments along with code listing)			
	5. Evidence of Testing (Test plan and associated screenshots**)**			

	6. Special Features (Describe any special features used in your code such as input validation, exception handling etc.)			
	7. Reflection Brief reflection on lessons learnt by completing this activity (~100 words)			

Part E **(5 marks)**	**1. Requirements Analysis:** (Describe the problem briefly in your own words (~100 words), for which you will develop a python software tool/solution)			
	2. Algorithm Design (either pseudo code or flowchart)			
	3. Hierarchy Chart / UML (UML if you have classes, and hierarchy chart if you have functions)			
	4. Brief Code Walkthrough – Console & GUI Program (Explanation of code fragments along with code listing)			
	5. Evidence of Testing (Test plan and associated screenshots**)**			
	6. Special Features (Describe any special features used in your code such as input validation, exception handling etc.)			
	7. Reflection Brief reflection on lessons learnt by completing this activity (~100 words)			

Total Mark (30): ……

Further Comments: