

## COMP.5202 Fundamentals of Programming & Problem Solving

---

Semester 2, 2017

---

Assessment	Type of Assessment	Due	Marks	Weighting
Assessment #1	Practical programming assignment	13th October 2017, 4pm	130 Marks	50%

---

### Learning outcomes Assessed:

---

1. Apply programming concepts and tools to system management tasks.
2. Demonstrate awareness of procedural and object oriented programming.
3. Apply principles of implementation (user testing, deployment).
4. Demonstrate programming using core logic and mathematical concepts such as problem solving methods, critical thinking, abstract reasoning; and systems thinking.

# Instructions

---

This is an individual assignment to be completed using visual studio code.

Please ask any questions about the assignment directly with your tutor or post it in the slack channel.

The assignment needs to be submitted using git. You will submit it using 2 links. One for Part A and one for Part B.

**NOTE:** This assessment contributes to 50% to your overall grade in COMP.5202

# Part A – Object oriented programming

---

## Instructions

- Read the scenario and program specifications for PART A below.
- Create a One page website in Visual Studio Code for each of the questions in Part A, based the specifications given for each.
- Ensure that you document (or comment) your code thoroughly (include a description of the program, the name of the developer and the version) and name your classes, methods, arrays and variables effectively.
- Note: If you cannot get some of your code to compile or if it causes your program to crash immediately, comment out that piece of code so that your program will still run.

## Scenario

Rotovegas Airways, a small local airline, has just purchased a computer for its new automated system.

You have been asked to develop a prototype of the new system. This will be developed in three parts. The first part is to be a program which will replicate the airline staff login system, the second is to replicate the airline seat booking system and the third one is to replicate the staff pay system.

# Question 1: Methods – Airline Login System

You are to write a single webpage to replicate the airline login system.

The initial screen should contain a welcome message and then prompt the user to set up their staff login by entering the following:

- a username (which must be 8 characters long)
- a password (which must be entered twice i.e. Pwd1 and Pwd2 and both entries must match)

All three of these (the username and two passwords) should be declared as static variables that are accessible by all methods in the program.

The program should contain two static methods as outline below:

- A ValidateUsername method that checks the username is at least 8 characters long
- A ValidatePassword method that checks both password entries match

Appropriate error messages should be displayed if the username and passwords do not meet the correct criteria, then giving user an opportunity to try again.

If all data is correct, the message “**Username and password has been set**” should be displayed.

## Question 2: One dimensional arrays – Airline Booking System

You are to code a single web page to assign seats on each flight of the airline's only plane (capacity: 10 seats).

- If the user enters 1, your program should assign a seat in the first-class section at the front of the plane (seats 1–5) and then tell the user which number seat they have been assigned to.
- If the user enters 2, your program should assign a seat in the economy section at the back of the plane (seats 6–10) and then tell the user which number seat they have been assigned to.
- Use a one-dimensional array of type Boolean to represent the seating status of the 10 seats on the plane. Initialize all the elements of the array to false to indicate that all the seats are empty.

As each seat is booked, set the corresponding element of the array to true to indicate that the seat is taken and no longer available. Your program should never assign a seat that has already been assigned.

Once the user has booked a seat, the program should ask if they want to book another seat, if they enter a 'Y' the program will loop letting them book further seats, and continue looping until they enter an 'N', they have no more bookings for that flight, then the program will close.

**EXTENSION:** When the economy section is full, your program should inform the user and ask them if they would consider a seat in the first-class section instead (and vice versa). If they respond with a 'Y', make the appropriate seat assignment. If 'N', then display the message "The next flight leaves in 3 hours."

## Question 3: Classes – Employee Class

You are to write a console application to replicate the Employee pay system for Rotovegas Airways.

In this program you are to add and code a class called Employee The class will include:

- **Three instance variables;** first name (type string), last name (type string) and salary (decimal).
- **Three set methods,** one for each of the instance variables listed above.
- **A `displayEmployee` method** which will display all of the employee's information.

In the Main method of the program you are to:

- Create an object of the Employee class
- Display a welcome message
- Prompt the user to input their first name, last name and annual salary (passing these to each of the set methods)
- Using the `displayEmployee` method, display all of the employee's information to the console screen.

# Part B – UPGRADE THE TEMPERATURE APP

---

- Read the scenario and program specifications for PART B below.
- Create a Single Web Page in Visual Studio Code, based the specifications outlined, determine the layout of the form and the objects or controls required on it.
- Ensure that you document (or comment) your code thoroughly (include a description of the program, the name of the developer and the version) and name your objects and variables effectively.
- Draw up a Flow diagram for this program. Create this in Draw.io and add it to your repository.
- **Draw up a Test Plan, which tests at least 3 scenarios. Create this in either word or PowerPoint. Make sure that it is included in the zip file you submit with your project.**

## Program Specifications:

For this part of the assignment you are to redevelop your temperature conversion program you did in the test (see test instructions over the page) but this time create it as a Single Web Site, using textboxes for your input and output.

- Use labels appropriately so it is clear what each textbox is for
- Change the properties of any textboxes that are not for input to be read only and not have a tab stop.
- Use a button click event to trigger the calculation.
- Display the name of the program in a **h1** tag centered above the form.

# WARNING

---

It is your responsibility to ensure that the correct source file is submitted, with the correct filename, and that it compiles without errors in a Toi Ohomai computer laboratory.

Failure to meet any of these requirements may mean that you lose marks for the operation of your program.

The assignment must be a product of your own work except for use of resources supplied with the course, discussions conducted by the lecturer during class time, and other assistance shown as acceptable in the section Assistance to Other Students.

NOTE: Your lecturer may use an oral examination to test your understanding of the material submitted.

---

## Assistance to other Students

Students themselves can be an excellent resource to assist the learning of fellow students, but there are issues that arise in assessments that relate to the type and amount of assistance given by students to other students. It is important to recognise what types of assistance are beneficial to another's learning and also what types of assistance are unacceptable in an assessment.



## Beneficial Assistance

- Study Groups.
- Discussion.
- Sharing reading material.
- Testing another student's programming work using the executable code and giving them the results of that testing.

## Unacceptable Assistance

- Working together on one copy of the assessment and submitting it as own work.
- Giving another student your work.
- Copying someone else's work. This includes work done by someone not on the course.
- Changing or correcting another student's work.
- Copying from books, Internet etc. and submitting it as own work. Anything taken directly from another source must be acknowledged correctly: Show the source alongside the quotation.

# Deliverables:

---

- You are to hand in 4 git repository links with the following titles:
  - studentid\_partA\_part\_1
  - studentid\_partA\_part\_2
  - studentid\_partA\_part\_3
  - studentid\_partB

**PLEASE FOLLOW THIS FORMAT SO IT IS EASY TO SEE WHO BELONGS TO WHAT PROJECT**

You are to submit all 4 repository links on moodle.