Deakin College
SIT105 Critical Thinking and Problem Solving for IT
Assignment 2 Trimester 1 2017

This assignment is to be completed individually. It is worth 25% of overall marks. It is due on or before 11.55 pm Sunday May 21.

Objectives:

There are several goals for the assignment:

- To demonstrate your understanding of some of the concepts behind programming;
- To demonstrate a methodical approach to solving problems; and
- To design and develop algorithms using sequence, selection and iterative constructs and modularisation.

Tasks:

- Carefully read all the questions and provide solutions to all four questions.
- Word process your solution and convert the whole document to **.pdf**
- Submit your assignment in the assignment submission tool available in Moodle (Week 11). Late submissions will be penalised. Note: The assignment is marked out of 50 marks but is worth 25 marks i.e. 25%
- Your assignment will be assessed on how well your solutions have addressed the problem and met the requirements, and on the overall "correctness" of your solutions. Use of appropriate constructs; appropriate messages/error messages; suitable formatting of output; and use of appropriate modularisation techniques are further assessment criteria.

1. Draw a defining diagram (IPO Chart) for the following problem statement (5 marks).

A program that reads ELEVEN numbers find their average and print it. The program should also print the number of times the number 6 occurs in the data. For example, given the input data:

4 6 9 6 5 6 10 7 0 16

The algorithm should print 7 as the average and 3 as the number of times 6 occurs.

2. Develop an algorithm for the following problem statement. Your solution should be pseudocode with appropriate comments (5 marks).

Using 1 mile = 1.85 kilometres, write an algorithm that will accept a value in miles as the speed measured by the car's speedometer. Convert that measurement to kilometres and alert the driver if he or she is travelling more than 90 km/hr.

3. **Develop an algorithm for the following problem statement. Your solution should be a pseudocode with appropriate comments. You should use modularisation in your solution and provide appropriate error messages (10 marks).**

A program that prompts the user for an annual salary and then calculate the amount of tax that should be paid. The tax rate that applies is shown in the table below.

Annual Salary	**Tax**
0 – $18,200	Nil
$18,201 – $37,000	19c for each $1 over $18,200
$37,001 – $80,000	$3,572 plus 32.5c for each $1 over $37,000
$80,001 – $180,000	$17,547 plus 37c for each $1 over $80,000
$180,001 and over	$54,547 plus 45c for each $1 over $180,000

If an invalid annual salary has been entered (negative or zero), then the user should be asked to re-enter the annual salary again. The annual salary and the amount of tax should be displayed to the screen with appropriate messages.

4. **Develop an algorithm for the following problem statement. Your solution should be a pseudocode with appropriate comments, a hierarchy chart and a defining diagram. You should use modularisation in your solution and provide appropriate error messages (30 marks).**

A program that accepts number of passengers on a bus and their age. The program should accept the fare based on the following:

- Age < 12 [Children] = AUD 10
- 12 < Age < 40 [Adult] = AUD 20
- Age > 40 [Senior] = AUD 15

The program should output the total fare for adults, children and seniors, and the grand total, which is, Adults + children + seniors.