ITECH5403 - Assignment 2 – Parallel Implementations

Due Date: 4pm, Friday of Week 11

This assignment will test your skills in programming applications to specification in a number of different programming languages, and is worth 20% of your non-invigilated (type A) marks for this course.

Assignment Overview

You are tasked with creating a program for a pizza shop – however, as this is a comparative languages course, you will be creating the same application in the following programming languages:

- C,
- Python,
- Java and
- Lisp.

As you implement the application in each language, you should keep notes on the features of the languages used, which you found useful, as well as any issues or complications which arose due to the complexity or lack of any language features. A brief discussion based on these programming features for each individual language accompanying each implementation is required. Finally, a comparative overview of the languages highlighting applicability based on your experience with the design, implementation and debugging of your code is also required.

If you foresee or encounter any complications, you may opt to implement or incorporate additional language features which may be lacking, i.e. data structures. This can be done via:

- Your own implementation,
- Through libraries, or
- Via the incorporation of existing source code. You can use code found on the Internet, but use of any existing code **must be referenced**.

Program Specification

Users may order one or more pizzas, where each pizza may be either: **small**, **medium** or **large**.

Small pizzas cost **$5**, medium pizzas cost **$8** and large pizzas cost **$12**.

All pizzas come on a **tomato base** (for our pizza shop, this will be the only option), and will have the topping **cheese** by default, at no extra cost. Users may choose up to a maximum of four additional toppings (bringing the total to five) from the following list, where each topping adds an additional $1 to the price of the pizza:

- Bacon,
- Olives,
- Ham,
- Mushrooms,

- Pineapple,
- Salami,
- Anchovies.

A pizza order consists of an order for one or more pizzas, where each pizza has a size, and may optionally include a list of up to four additional toppings.

Each pizza order must be marked as either to be **collected** or to be **delivered**.

If the pizza is to be collected then the order requires a **name** and a **phone number** to be valid.

If the pizza is to be delivered then a **name**, **phone number** and **address** are required to be valid. In addition, if the order total is less than $30 then an $8 delivery fee is added to the total.

The application must be error tolerant and capable of accepting keyboard input to store a number of pizza orders in memory (they do not have to be persisted to file), as well as displaying an order summary which include details of all orders, including:

- The details of each pizza in the order,
- The total cost of the order, and
- The name, phone number and (if required) address of the person who made the order.

Suggested Development Environments

Codeblocks for C '99

Code::Blocks can be downloaded from: http://www.codeblocks.org/downloads/binaries

To create a new C project is: When you create a project, choose **File | New** and then **Console Application**, and then choose **C** as the programming language.

IDLE for Python

Python, including the IDLE development environment can be downloaded from:
https://www.python.org/downloads/

Eclipse for Java 7 or Java 8

Eclipse may be freely downloaded from: http://www.eclipse.org/downloads/

Eclipse does not come with the Java JDK, which must the downloaded separately from:
http://www.oracle.com/technetwork/java/javase/downloads/index.html

Ensure that your Eclipse type and Java type match – i.e. 32-bit Java for 32-bit Eclipse, or 64-bit Java for 64-bit Eclipse. If you mix and match it won't work.

GNU CLisp for Common Lisp

CLISP 2.49 can be sourced from: http://sourceforge.net/projects/clisp/files/latest/download

Any good text editor would be suitable for writing the source code.

Additional Documentation – Language Suitability Report

The design of each programming language incorporates a number of decisions about the language which make it more or less suitable for given tasks. During your implementation of the pizza program in each of the languages you should make notes about the language features which exist or do not exist, and which have therefore made program development easier or more difficult.

Where a language has not provided a feature which would have been useful to the implementation of the program, or where the complexity of using a language feature has been high you should remark upon it and briefly discuss a mechanism or feature of another language which would have made development easier.

After completing the application in all languages (or as many as you can), discuss the comparative ease of implementation in terms of the design, implementation and debugging for each programming language, including how robustness issues were addressed.

Submission and Marking Process

You must supply your program source code files and language suitability report documentation in as single compressed archive called:

ITECH5403_Assignment_2_<YOUR-NAME>_<YOUR-STUDENT-ID>.zip

You may supply your programming language suitability report in either Word or LibreOffice/OpenOffice format in which the document can be edited – no proprietary Mac specific formats, please.

Assignments will be marked on the basis of fulfilment of the requirements and the quality of the work. In addition to the marking criteria, marks may be deducted for failure to comply with the assignment requirements, including (but not limited to):

- Incomplete implementation(s), and
- Incomplete submissions (e.g. missing files), and
- Poor spelling and grammar.

Submit your assignment (all program source files plus your discussion document) to the Assignment 2 Upload location on Moodle before the deadline of Friday of Week 11 at 4pm.

The mark distribution for this assignment is explained on the next page.

Assignment 2 – Parallel Implementations

Student name: **Student ID:**

Requirement	Weight	Mark
Implementation of the pizza shop program in the C programming language. Areas of note include: - Use of data structures, - Robust input handling which does not cause program termination if provided with bad data (i.e. program expects a number, gets given alphanumerical data). Discussion on implementation: - Language features, issues and suitability.	15 5	
Implementation of the pizza shop program in the Python programming language. Areas of note include: - Python Standard library, - List mechanisms, Discussion on implementation: - Language features, issues and suitability	15 5	
Implementation of the pizza shop program in the Java programming language. Areas of note include: - Object orientation mechanism / method calls, - Error handling - Standard Java libraries Discussion on implementation - Language features, issues and suitability	15 5	
Implementation of the pizza shop program in the Lisp programming language. Areas of note include the Lisp: - Use of recursion - lists - Inbuilt data structures Discussion on implementation - Language features, issues and suitability	15 5	
Documentation and discussion of the comparative ease of implementation (design / implement / debug) in each programming language, including how robustness issues were addressed.	15	
Spelling and grammar	5	
Assignment mark total		**/ 100**
Contribution to unit mark (out of 20%)		**%**

Comments: