****

Software Development: Object Oriented Programming

(H171 35)

The Unit Project

“A version of the Sokoban game”

Completion of this open book practical assessment satisfies the following unit requirements:

Outcome 1

”To investigate object oriented programming techniques and apply them to a design”

Outcome 2

“To implement a solution from an object oriented design using object oriented techniques”

Outcome 3

”To test the completed product”

The Project Brief

Refer to the section below entitled

“The scenario: A reminder of your client’s brief”

Your task, as the solution provider

This project can be carried out under supervised and unsupervised conditions

i.e. you may work on it in your own time

**Note that although this is an open book assessment, assessors will NOT accept anything other than what they consider to be a student’s own submission**

**Note also that, to mimic a real life requirement, the deadline date provided is not negotiable (except under extenuating circumstances)**

Deadline dates:

Implementation deadline: 22/2/18

Testing deadline: 8/3/18

You are currently creating the required analysis and detailed design documentation (geared towards an OOP implementation) based on the Sokoban game project brief provided as the instrument of assessment for the “Systems Development: Object Oriented Analysis and Design” (H172 35) unit

(Outcomes 2 and 3)

You will now continue with the full development life cycle of the Sokoban game solution by implementing the solution in Visual C# and testing your solution

The scenario

A reminder of your client’s brief

Retro Games Ltd. has commissioned you to design and develop a simplified version of Sokoban

Sokoban is a popular single player game by which a player pushes crates around a map to get them all in the right location

The game is played on a 2 dimensional grid, but the rooms are not usually of regular shape

The edges of the room are indicated by a wall, and the player and boxes cannot get through the wall

There is a warehouse keeper, who the player must control in order to move the crates from their starting positions onto the diamonds

The diamonds are the end points for the crates

You can only push a crate when you are to one side of it and its opposite side is clear, which makes the task somewhat tricky for more complicated maps

There are walls all around the map, and also in the middle in various configurations

Crates cannot be pushed through walls

Once a crate is up against a wall you can only push it along the wall, as you need to get behind a crate in order to push it

Once a crate is in a corner it is impossible to move it again

The warehouse keeper is unable to climb over crates, and is only strong enough to move one crate at a time

Crates can only be pushed, not pulled

The game will require at least five levels

Each level is to be harder to solve than the previous one, either by having more crates or obstacles, or tighter corridors, or a more complex starting arrangement of crates

The program is to record how many moves a player takes to solve a level, and output this information visually

This unit project consists of two stages:

* Stage 1 is the program implementation
* Stage 2 is the testing of the completed program

Read the Evidence Requirements stated below for each stage and clarify any points with your assessor before you commence the project

Evidence Requirements for Stage 1: Implementation

(covering Outcomes 1 and 2)

You are required to implement the program in an object oriented manner

The solution must demonstrate each of the following:

Design requirements:

* abstraction, encapsulation and information hiding used where appropriate
* inheritance used if appropriate to the solution
* polymorphism used if appropriate to the solution
* all class-wide variables are private to prevent content coupling
* class-wide variables are kept to a minimum to ensure a minimum of common coupling
* data coupling is used (parameter passing) in preference to content or common coupling
* program does not contain a lot of unnecessary data coupling
* classes are highly cohesive

Implementation requirements:

* a working solution which meets the requirements of the given brief
* variables are correctly declared and initialised
* arithmetic and/or logical operators are used correctly
* a range of control structures are implemented correctly
* at least two data structures are implemented correctly
* the program contains a minimum of four classes, which contain attributes, methods and a constructor method
* a minimum of three objects are created from the classes, with appropriate initial attribute values set through the constructor methods
* the program contains at least one overloaded method (this may be a constructor method)
* classes are linked appropriately through association, aggregation or inheritance relationships
* parameters are passed correctly both within and between objects
* appropriate access types are defined for methods, attributes and classes
* use of pre-defined classes and/or methods from the standard object library
* the program appropriately handles errors with exceptions or pre-validation
* the program code is commented appropriately throughout

Evidence Requirements for Stage 2: Testing

(covering Outcome 3)

After completing Stage 1 you are required to develop a test plan and test your solution

The test logs should identify any areas where the program fails, and detail any fixes and retests required

Additional information

Your assessor will check the authenticity of any work you have done unsupervised

This may involve methods such as interviews, demonstrations, checking files, etc. and may be carried out at random and pre-arranged times

Your assessor will specify the various deadline periods for the project

It is up to you to determine your own deadlines within these

You may decide to work on multiple tasks at the same time but you should try to fully complete and achieve one stage before completing the next

Applying this method of working is good preparation for the

“Computing: Software Development: Graded Unit 2” (H48W 35)