Software Design & Development – Revision Questions 1: Answers

1	The condition at line 2 would be evaluated as TRUE because 12 is greater than 10. This would cause the error message at line 3 to be displayed and the user prompted to re-enter at line 4. The loop would return to line 2 and the condition checked again.
2	**0, -1, -2** and below **901, 902, 903** and above **Banana, Fred, abc**… *Exceptional data is anything out with the acceptable inputs.*
3	**Syntax Error** *This is because the rules of the programming language have been broken. UNTOL is not a word used in the programming language.*
4	Programs should be readable so that they can be easily understood. This will help other programmers to identify what the program does, the purpose of variables and the beginning and end of construct (Selection, Iteration)
5a	**Line 3 contains a logic error** *This is because the condition accepts either a valid username or a valid pin number but both don't have to be correct.*
5b	**Change OR to AND** *This will make the condition require both the username and the pin number to be correct in order for it to be TRUE.*
6a	**Input Validation** *Input validation is used to check that the values entered by the user are acceptable, and asks them to re-enter if they are not. That is what is happening in this code.*
6b	0.05 **Exceptional** *(out with the range of 0.10 and 0.50)* 0.45 Normal 0.10 **Extreme** *(on the lower boundary of acceptable data)* **0.50** Extreme *(on the upper boundary of acceptable data)*
7	**Normal: 2, 3, 4…** up to **…77, 78, 79** **Exceptional: 0, -1, -2** and below **/ 81, 82, 83** and above **/** Banana
8	The conditional loop starts at line 5 The user is prompted at line 6 The user enters a password at line 7 If the condition at line 8 is TRUE then the loop terminates. If the condition is false, the loop returns to line 5 and the user is asked to re-enter the password.
9	The condition at line 2 would be evaluated as TRUE because 63 is less than 82. This would cause the error message at line 3 to be displayed and the temperature detected again at line 4. The loop would return to line 2 and the condition checked again.

10	Badgename: **Array / String** Workgroup: **Array / Integer** *The data structure is array for each because they are storing multiple items (list)* *Badgename is string because it is storing letters* *Workgroup is Integer because it is storing whole numbers*
11	Topscores: **Array / String** *The data structure is array because it is storing multiple items (list)* *Topscores is Integer because it is storing whole numbers*
12a	**Allows the programmer to identify the beginning and end of iteration and selection constructs (as well as the contents of these)**
12b	**Internal commentary** **Meaningful variable names** **Use of white space**
13	**Internal commentary** **Meaningful variable names** **Use of white space** **Use of indentation**
14ai	**Flow Chart** **Structure Diagram**
14aii	**These are both graphical design notations which provide a visual representation of the program design which could make it easier to follow.**
14b	**Popularity: Integer –***Storing whole numbers* **Weblink: String -** *Storing letters* **Shares: Integer -** *Storing whole numbers*

14c	**Around Line 4 insert:**
	IF shares > -1 THEN
	SET popularity TO popularity + shares
	END IF
	
	When the user enters -1 at line 3 to stop the program, the -1 is being added to popularity at line4 before the condition to terminate the loop is checked at line 5.
	
	By placing a selection construct (IF) around line 4, the -1 will not be added to the popularity because it is not > -1. The loop will then terminate at line 5.

15a	**Between lines 4 and 5 insert:** **IF total_time =45 THEN** **SEND ("Warning") TO DISPLAY** **END IF** *By adding a selection construct (IF) after line 4, each time the total_time is updated the condition will be checked to see if it is equal to 45. When the condition is true, the warning message will be displayed.*

15b

```
IF passcode(1) = usercode(1) AND passcode(2) = usercode(2) AND passcode(3) = usercode(3) AND passcode(4) = usercode(4)
    SEND “unlocked” TO DISPLAY
END IF
```

OR

```
REPEAT index FROM 1 to 4
    IF passcode(index) = usercode(index)
        SET counter TO counter + 1
    END IF
END REPEAT
IF counter = 4 THEN
    SEND “unlocked” TO DISPLAY
END IF
```

Each position in the passcode array must be compared with each position in the usercode array. If all four passcode positions match the usercode positions then the usercode is correct and the unlocked message can be displayed.

15c

There is no need because the user interface only allows the values 1 to 9 to be selected for each position.
It is impossible to enter an invalid value so no validation is needed.

16a

Conditional loop

The loop does not repeat a set number of times, it repeats until a condition is true (counter is over 0) so it is a conditional loop.

16b

Syntax Error

The rules of the programming language have been broken and the program cannot run. ; is not expected by the programming language in this case.

17a

```
SET total TO 0
REPEAT index FROM 1 TO 30
    SET total TO total + hours(index)
END REPEAT
SEND total TO DISPLAY
```

This program uses a fixed loop to run through each position in the hours array. A running total is used to add the hours in each day as the loop runs through the array.

17b

At line 3 the complex condition will be evaluated as TRUE because 15 is more than 12.
This will cause the error message to display at line 4.
The complex condition at line 6 will be evaluated as FALSE and so the loop will repeat, asking the user to re-enter at line 2.

17c

Conditional loop

The loop does not repeat a set number of times, it repeats until a condition is true so it is a conditional loop.

18a	**Line 4 – OR** should be **AND** **Line 8 –** should be **UNTIL temperature <= 20 AND toxic_level <=50** *Line 4 is wrong because the vents should only open if both the temperature and the toxic level are too high. In this case, they will open if only one of them is too high.* *Line 8 should repeat until the temperature and the toxic level return to normal. In this case, the loop will stop if the temperature is over 20.*
18b	**Temperature = 20, toxic level = 20 : normal** **Temperature = 20, toxic level = 50 : extreme -** *both values which make the loop start* **Temperature = 7, toxic level = 34 : normal –** *both reasonable values* **Temperature = 10000, toxic level = Banana : Exceptional –** *impossible values*
19a	**Line 4 – Both N and E have to be touching to make it turn left. The OR should be changed to an AND** **Lines 6 to 13 – This code is contained inside the condition at line 4, so to turn right it must be touching N and E and then also W. These lines should be moved to below line 13 (outside the first condition)**
19bi	**Operation = 27, power= 2 : normal** **Operation = 60, power= 1 : extreme -** *both values on the boundary* **Operation = 65, power= 4 : exceptional -** *both values out with the boundary* **Operation =1 , power= 3 : extreme -** *both values on the boundary*
19bii	**A program should be fully tested to ensure that it works for all possible inputs without crashing.**