Software Design & Development – Revision Questions 1

1.

This pseudocode allows a user to enter the level they wish to start playing a game.

```
Line 1    RECEIVE level FROM (INTEGER) KEYBOARD
Line 2    WHILE level < 1 OR level > 10 DO
Line 3    SEND “error : please re-enter level” TO DISPLAY
Line 4    RECEIVE level FROM (INTEGER) KEYBOARD
Line 5    END WHILE
```

Explain what happens if a user enters 12.

2.

A programmer is developing a stock control program. If a user enters a stock code number from 1 to 900, it will display the number of items in stock.

Give one example of exceptional test data the programmer could use to test the program.

3.

When the code for a conditional loop is written the programmer mis-types the word UNTIL, typing UNTOL instead.
State the type of programming error being described above.

4.

Explain why it is important that program code is readable.

5.

Employees can only access their company network if they enter a correct username and password. A validation program is being developed and will run each time an employee logs on.

An extract of pseudocode from the program is shown below.

```
Line 1    RECEIVE userName FROM (STRING) KEYBOARD
Line 2    RECEIVE pinNumber FROM (STRING) KEYBOARD
Line 3    IF userName VALID OR pinNumber VALID THEN
Line 4        Allow access to network
Line 5    ELSE
Line 6        SEND "Access Denied" TO SCREEN
Line 7    END IF
```

(a) Identify the line containing a logic error.

(b) State how this error should be corrected.

6.

Jack writes a program to ensure that the cost of water can only be entered as always greater than or equal to £0·10 and less than or equal to £0·50 per litre.

(a) State the standard algorithm that is used to ensure that data entered is acceptable.

(b) Copy and complete the table below to show four different examples of test data for water cost and the type of each example.

Test Data	Type of Test Data
0·05	
0·45	normal
0·10	
	extreme

7.

This pseudocode allows the user to guess the age of a teddy bear to win it in a competition.

```
Line 1      RECEIVE guess FROM (INTEGER) KEYBOARD
Line 2      WHILE guess < 1 OR guess > 80 DO
Line 3          SEND “invalid guess: please try again” TO DISPLAY
Line 4          RECEIVE guess FROM (INTEGER) KEYBOARD
Line 5      END WHILE
```

Give one example of **normal** test date and one example of **exceptional** test data for guess.

8.

Explain the purpose of lines 5 to 8 in this pseudocode.

```
…
Line 4      SET password TO “h1gh@sch00l”
Line 5          REPEAT
Line 6 SEND “Please enter your password” TO DISPLAY
Line 7      RECEIVE user_guess FROM (INTEGER) KEYBOARD
Line 8      UNTIL password = user_guess
```

9.

This code design monitors the temperature of food as it is reheated.

```
Line 1      RECEIVE temperature FROM (REAL) <temperature sensor>
Line 2      WHILE temperature < 82 DO
Line 3      SEND “temperature too low: continue to reheat” TO DISPLAY
Line 4      RECEIVE temperature FROM (REAL) <temperature sensor>
Line 5      END WHILE
```

Explain what will happen in lines 2 to 5 if the sensor detects 63°.

10. The pseudocode below shows how a program creates a number of labels for ID badges.

```
Line 1.   SET badgename TO [“Charlie Burton”, “Colin Shaw”, “Morag Burke”,
                                “Gordon Chisholm”]

Line 2.  SET workgroup TO [31, 67, 12, 19]
```

State the most suitable data structure and data type for storing the both badgename and workgroup.

	Data Structure	Data Type
badgename		
workgroup		

11. The pseudocode below shows how a program could store and process the top scores of the players in an online computer game.

```
Line 1.     SET topscores TO [1209, 1302, 1532, 1092, 1892, 1365,
                                2341, 2321, 1539 1155]
Line 2.     SET high_score TO topscores [0]
Line 3.     FOREACH score FROM topscores DO
Line 4.         IF score > high_score THEN
Line 5.             SET high_score TO score
Line 6.         END IF
Line 7.     END FOREACH
Line 8.     SEND [“The top score is: “, high_score] TO DISPLAY
```

State the most suitable data structure and data type for storing the highlighted variable (topscores) used above.

12. Below is a section of code written in the programming language Java.

```
public int[] selectionSort(int[] data){
  int x = data.length;
  int j = 0;
  int tmp = 0;
  for(int i=0;i<x;i++){
    j = i;
    for(int k = i;k<x;k++){
      if(data[j]>data[k]){
        j = k;
      }
    }
    tmp = data[i];
    data[i] = data[j];
    data[j] = tmp;
  }
  return data;
}
```

(a) This code makes use of indentation. Explain why indentation improves the readability of code.

(b) State two other techniques that a programmer could use to make this code more readable.

13. The following section of code is written in the programming language Ada.

```
while not Text_IO.End_Of_File loop
while not Text_IO.End_Of_Line loop
Text_IO.Get (Item=>Char);
if Char in Printable then
Counts (Char) := Counts (Char) + 1;
end if;
end loop;
Text_IO.Skip_Line;
end loop;
for I in Counts
if Counts(I) > 0 then
Text_IO.Put (I & ":  ");
for J in 1 .. Counts(I) loop
Text_IO.Put ('*');
end loop;
Text_IO.New_Line;
end if;
end loop;
```

State two techniques that the programmer could use to make this code more readable.

14. Social media allows users to share web links with others.

Shares for http://www.glew.org.uk	
Social Network	**Number of Shares**
facebook	236
Google+	231
tumblr.	190
twitter	410

To calculate the popularity of a web link, the number of times it has been shared on different social media platforms are added together.

In the table shown, the popularity is 1067 (236 + 231 + 190 + 410)

A program is designed to calculate the popularity of a web link from a list of shares on social media sites.

```
Line 1.   SET popularity TO 0
Line 2.   RECEIVE weblink FROM keyboard
Line 3.   SET weblink TO lowercase(weblink)
Line 4.   REPEAT
Line 5.       RECEIVE shares FROM keyboard
Line 6.       SET popularity TO popularity + shares
Line 7.   UNTIL shares = -1
Line 8.   SEND ["The popularity of web link "& weblink & " is "&
              popularity & "."] TO DISPLAY
```

(a)

(i) The above design was created using pseudocode. Name another design notation that could have been used instead.

(ii) Describe one advantage of using this design notation rather than pseudocode.

(b) Identify the variables and state their data types used in the program design.

(c)

```
Line 1.     SET popularity TO 0
Line 2.     REPEAT
Line 3.         RECEIVE shares FROM keyboard
Line 4.         SET popularity TO popularity + shares
Line 5.     UNTIL shares =  -1
```

The program above stops when the user enters -1.

There is an error in this program which means that -1 is always added to the "popularity" before the program finishes.

Use pseudocode or a programming language of your choice to show how you would correct the program so that this error is removed.

15. A countdown timer on a website is created to give a user 60 seconds to enter his or her passcode before it automatically blocks the user from access the site.

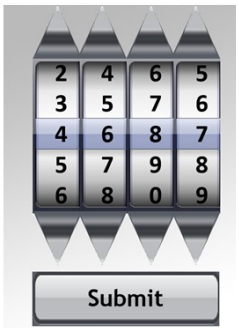
The code for the timer is shown below.

```
Line 1.     SET time to 0
Line 2.     REPEAT
Line 3.         WAIT 1 seconds
Line 4.         SET total_time to total_time +1
Line 5.     UNTIL total_time = 60
Line 6.     SEND ["Time to enter has now expired"] to DISPLAY
```

(a) The program stops when time is equal to 60. The design is changed to include a warning message when the time is more than 45.

Use pseudocode or a programming language you are familiar with to show how this additional feature could be implemented.

(b) This screenshot shows the interface used to enter a user's passcode.



When the submit button is clicked the highlighted numbers are read into an array called usercode. This has to be matched with the passcode which is stored in an array called passcode.

Write, using pseudocode or a language you are familiar with, code that would display the message “Unlocked” when the array elements in ‘usercode’ are equal to the equivalent elements in ‘passcode’.

(c) Explain why exceptional test data will not be required to test the passcode screen program.

16.

(a) State the type of loop shown in the program below. Justify your answer.

```
SET newstring TO “computing”
SET counter TO 3
WHILE counter > 0
    SET newstring TO swap(counter, counter + 2, newstring)
    SET counter TO counter – 1
END WHILE
echo(newstring)
```

(b) A change is made to the program but when it is translated the following error message is displayed.

Error: Unexpected use of “;” at line 12

Name the type of error that has occurred. Justify your answer.

17. Paul is a student and has written a program to record how much he studies each day in the 30 days before his exams start. Each day he enters a value for the total number of hours used for study that day. Each of these values is stored in an array called 'hours'.

(a) Using pseudocode or a programming language of your choice, write a short program which will tell Paul the average number of hours he studied over the 30 days.

(b) The pseudocode shown below shows how the hours are entered.

```
Line 1:    REPEAT
Line 2:        RECEIVE study_time FORM keyboard
Line 3:            IF study_time < 0 OR study_time > 12 THEN
Line 4:                SEND error message to display
Line 5:            END IF
Line 6:    UNTIL NOT (study_time < 0 OR study_time > 12)
```

Describe all of the events that will occur if Paul enters the value 15.

(c) State the type of loop shown in the program above. Justify your answer.

18. A factory is equipped with sensors. These sensors use embedded technology to monitor levels of toxic gas in areas of the factory.

If the temperature in an area is above 20 degrees and the toxic level is above 50 then the vents are opened in that area of factory.

The pseudocode below shows the design for the program. There are two errors in the logic of the program. Find and describe each error.

(a) The pseudocode below shows the design for the program. There are two errors in the logic of the program. Find and describe each error.

```
Line 1: RECEIVE temperature FROM real sensor
Line 2: RECEIVE toxic_level FROM real sensor
Line 3: RECEIVE factory_area FROM keyboard
Line 4: IF temperature > 20 OR toxic_level >50 THEN
Line 5:     REPEAT
Line 6:         SEND open vents TO vents
Line 7:         SEND [ factory_area & " has vents open"] to DISPLAY
Line 8:     UNTIL temperature > 20
Line 9: END IF
```

(b) The program is tested using the following test data.

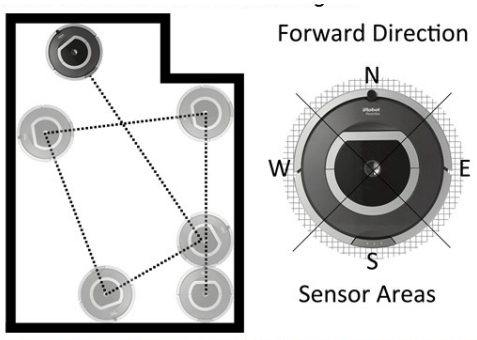
Complete the table below to show four examples of test data and the type of each example.

Test Data	Type of Test Data
temperature = 20, toxic_level = 20	Normal
temperature = 20, toxic_level = 50	
temperature = 7, toxic_level = 34	
	Exceptional

19.

A robot uses an algorithm to decide how it should act when moving around a room. The robot can only move forward in the direction shown. The robot has four "Sensor Areas" and when any of these hit an obstacle the robot will respond.

If it detects contact in areas "N" or "E" it turns left. If there is contact in area "W" it turns right. If the rear of the robot is bumped it stops, waits 10 seconds and then moves forward again.



(a) The pseudocode below shows a design for part of the program used by the robot. There are two errors in the logic of the program design. Find and describe each error made.

```
Line 1.     REPEAT
Line 2.         SEND move forward TO motor
Line 3.         RECEIVE touch FROM (string) SENSOR
Line 4.         IF touch = "N" AND touch = "E" THEN
Line 5.             SEND turn left TO motor
Line 6.             IF touch = "W" THEN
Line 7.                 SEND turn right TO motor
Line 8.             ELSE
Line 9.                 SEND stop TO motor
Line 10.                WAIT 10 Seconds
Line 11.                SEND move forward TO motor
Line 12.            ENDIF
Line 13.        ENDIF
Line 14.    UNTIL end of cleaning session
```

(b)

To operate the robot two settings are entered. The length of operation, in minutes from (1 to 60) and the power setting (where 1 is economy mode, 2 regular mode and 3 is super clean).

(i) Complete the table below to show the four examples of test data and the type of each example.

Test data	Type of test data
operation - 27 minutes, power - 2	normal
operation - 60 minutes, power - 1	
operation - 65 minutes, power - 4	
	extreme

(ii) Explain the purpose of fully testing a program using a variety of test data.