2022 ASSESSMENT REPORT
ITC315118 - COMPUTER SCIENCE

SECTION A

QUESTION 1

a) $550

b) $1500

c) 29 set VideoPrice to 1000

d) after line 34 (inside the else condition):

```
34a     if processor = 'n' and video = 'd'
34b         Display "Warning – you should buy a power processor"
```

Examiner Comments

- Items (a) and (b) were very well done by students.
- Item (c) Some students were confused about what "an increase of 100%" meant. Otherwise, question (c) was well handled, with there being a variety of correct answers given.
- Item (d) Most students added their solutions in the "When the calculate Button is pressed" and gained full marks. Some students put their answer in one of the other sections and did not consider the possibility that not all the data had been entered yet. These students received part marks for their answers.

QUESTION 2

a) Add after the line "speed = speed + 10":

```
if speed > 110
        speed = 40
```

b) When "speed" button is pressed

```
if user == novice

    speed = speed + 5

        if speed > 65

            speed = 40
```

c) Modify 'U': set spin to 'F'
to be: 'U': set spin to 'R'

and add: 'R': set spin to 'F'

Examiner Comments

- Question 2 (a) As this question did not have line numbers for the algorithm, none were expected in students' solutions. Almost all students gave a correct answer to this part.
- Item b) was also well done. Some students referred to their answer in a), then adding the extra case, others completely rewrote "When "speed" button is pressed". Both options were given full marks.
- Item (c) Most students did not realise that the bike had a single button that was to cycle through 3 options, now 4. The fact that the extra option was a "random" option made many students believe one of the other three options needed to be chosen at random. Answers were given where it would not be possible to reach some options by repeatedly pressing the button. Only a few students received full marks for this question.

QUESTION 3

Initially

player1Score = 501

player2Score = 501

dartValue = 0, dartsTotal = 0

dartType = single

dartsLeft = 3

currentPlayer = 1

currentScore = 0

Display 3 in DartsLeft TextField

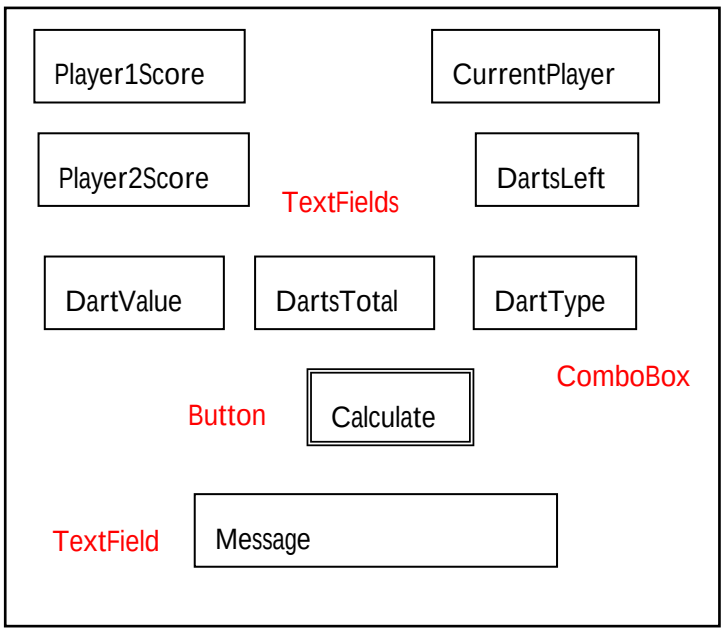
Display 0 in DartsTotal TextField

Display 0 in DartValue TextField

Display 501 in Player1Score TextField

Display 501 in Player2Score TextField

Display 1 in CurrentPlayer TextField



When a value is entered in dartValueTextField

```
Set dartValue to value in dartValueTextField
if dartValue > 20
        dartValue = 20
if dartValue < 1
        dartValue = 1
```

When a value is entered in dartTypeComboBox

```
Set dartType to value in dartTypeComboBox
```

When the "Calculate" button is pressed

```
if currentPlayer == 1
        currentScore = player1Score
else
        currentScore = player2Score
case dartType of:
        "IR":       dartsTotal = dartsTotal + 50
        "OR":       dartsTotal = dartsTotal + 25
        "triple":   dartsTotal = dartsTotal + dartValue * 3
        "double":   dartsTotal = dartsTotal + dartValue * 2
        "single":   dartsTotal = dartsTotal + dartValue
        "miss": dartValue = 0
if (currentScore – dartsTotal) == 0 and (dartType == "double" or dartType == "IR" or
dartType == "OR")
        Display "Player " currentPlayer " wins!" in Message TextBox
        Halt
if (currentScore – dartsTotal) == 1  or (currentScore – dartsTotal) < 0
        Display "Player " CurrentPlayer " is bust." In Message TextBox
        currentPlayer = currentPlayer % 2 + 1
        dartsLeft = 3
        dartsTotal = 0
if (currentScore – dartsTotal) > 1
        Display dartsTotal in DartsTotal TextBox
        dartsLeft = dartsLeft – 1
        if dartsLeft == 0
                currentScore = currentScore - dartsTotal
                dartsLeft = 3
                if currentPlayer == 1
                        Display currentScore in Player1Score TextBox
        else
                Display currentScore in Player2Score TextBox
        currentPlayer = currentPlayer % 2 + 1
        dartsTotal = 0
        Display 0 in DartsTotal TextBox
Display dartsleft in DartsLeft TextBox
Display currentPlayer in CurrentPlayer TextBox
```

Examiner Comments

The algorithm expected from this question was extremely difficult to perfect. Students were also hindered by a lack of an interface for them to scaffold from as well as an incomplete specification of the problem, i.e., what happens if a dart takes your score to 0, but you did not throw a double or a bullseye? Note that the algorithm above does not deal with that case either.

In marking this question no-one was expected to achieve perfection, though a few students came close. Instead, part marks were given for demonstration of each of the following concepts:

- Checking the dart value was in the range 1-20 (only a couple of students did this and it was not clear if it was required).
- Including a mechanism for changing between player 1 and player 2 and back again.
- Including a count of the darts in each turn.
- Correctly checking if a player went bust.
- Correctly checking if a player had won.
- Correctly calculating the score from an individual dart.

Many students achieved enough marks to take them to an 'A' rating for this section. Many more students were able to justify a 'B' rating by completing some of this algorithm. 20% of students did not attempt this question at all.

SECTION B

QUESTION 4

a) i. **y** 5.666 8.0 8 11

z 5.666 8.0 8 11

ii. 13 13.0 14 14.0

iii. 1 2 4 5

b) i. Final value of **d:** **false**

ii. Final value of **e:** **10**

Explanation: e (5) >0 therefore e will become equal to 10. e (10) is not greater than 10 so the second if statement has no effect.

iii. Final value of **f: 20**

i	f
1	2
2	6
3	12
4	20

iv. Final value of d**: 12**

y	d
0	6
3	12
6	

c) public double question4c (double cost)

// the value "input" is the value passed into the method.

```
{
        double output;
        output = cost / 11;

        return output;
}
```

Examiner Comments

a) Completed correctly by the majority of students, very few omitted an answer for each question.

b) Most students successfully completed items (i), (ii) and (iv). Only half managed to successfully complete item (iii). The issue seemed to be around the inability to evaluate the statement within the for loop.

c) Only half managed to complete this item successfully. Those that didn't, failed to understand the requirement to simply assign the variable output a value of cost / 11 or did not attempt the question at all. This was disappointing as the last line of the question gave students what the required expression was.

QUESTION 5

a)

i	k	data		
		0	1	2
0	2	2	5	1
1	9			
2	15			

Final value of **k**: **15**

b) String b;

b= part5b(1,2,3);

c) i. Final value of **m:**

m	0	1	2	3
0	0	2	3	4
1	1	3	3	4
2	1	2	6	4

ii. This would create an arrayOutOfBounds exception error within the inside loop as the loop counts to 4 but the highest index allowed is 3.

Examiner Comments

a) Completed correctly by around half of the students. Those that knew how to trace a nested for loop had little difficulty in completing the trace table and producing the correct final value for k. The remaining students produced a variety of different outputs which demonstrated a lack of understanding of this construct.

b) Only a third of students were able to complete this question successfully. About one sixth attempted to call the procedure but failed to use the correct syntax to call a method with a return value. About half the students did not attempt or were too far off the mark.

c) Completed correctly by a little under half of the students. Some students were unable to successfully complete the table. Many students identified that the array was no longer the correct size but were unable to explain the error that would occur, in this case partial marks were awarded. Many referred to overflow errors.

QUESTION 6

a) **sumx** = 371.0
sumy = 155.0

b) Primitive parameters (double in this case) are called by value only so when the method call is complete, any changes which have occurred will not be made to the calling parameters. This means that sumx and sumy will be incorrect and the overall results will also be incorrect.

c)

i. Code will still work but the for loop at line 21 will only process the first 7 elements, thus producing an incorrect result.

ii. Change the loop to for(int i = 0; i < **a.length**; i++)

Examiner Comments

a) Attempted by a little under half of the students, half of this group gaining full marks. The other half did not recognise that the variables, sumx and sumy, were declared as doubles and hence required 371.0 and 155.0, not 371 and 155.

b) This was the hardest item in the section. Only 7% of students were able to gain full marks for this question by explaining why the code would not work, making reference to the way in which primitive parameters are passed. A similar number attempted to explain and were given partial marks. Three quarters of students did not attempt this question.

c) It was pleasing that about 40% of students attempted this item successfully. Most correctly identified why the correct result would not be achieved and provided a solution to rectify. The more elegant solution was to use i < a.length in the loop rather than increasing 7 to 8.

SECTION C

QUESTION 7

a) ClimateControl c = new ClimateControl (27.0, 3, false);

b) 0

c) Double

d)
```
public void setTempDownByTen() {
        if (temperature > 10)
                temperature = temperature - 10;
 }
```

Examiner Comments

- Item (a). Around 30% of students incorrectly chose the answer with the String "true" instead of the boolean `false` for the 3rd argument.
- Most students gave the correct answer for (b) and (c).
- Item (d). Most students had a reasonable answer. About half the students with an otherwise correct answer used the wrong limit, with most of those copying the `<0` limit from the provided code.

QUESTION 8

a) Coins coin = new Coins( ) ;

b) coin.enter(500.0, 700.0, "Australia", "1958");

c)
```
public void setValue ( int num , double value) {
                        collectVal[num] = value;
    }
```

d)
```
double total;
for (int i = 0; i++ ; i<coin.getNumber() ) {
    total = total + coin.getCollect[ i ];
}
```

Examiner Comments

- Items (a) and (b). Generally, a wide range of answers and overall, not answered particularly well.
- Item (c). Of the students that otherwise answered this question well, about half missed that the year was supposed to be a String.
- Item (d). Some students were confused about how to use `getNumber()` to help with answering this question.

QUESTION 9

```
public class ElectricCar {
    String location;
     String address;
     int capacity;
     int quantity;
     String  plugTypes;
```

```
    int nearestTypeOne;
    int nearestTypeTwo;

public class ElectricCar(String loc, String addr, int cap, int qty, String  plug, int nOne, int
nTwo)  {

        location = loc;
        address = addr;
                        capacity = cap;
                        quantity = qty
                plugTypes = plug;
                        nearestTypeOne = nOne;
                nearestTypeTwo = nTwo;

    }

        public String info ( ) {
                return location + ", " + capacity + ", " + plugTypes;
         }

         public boolean withinRange(int myRange, String myPlugType) {
          boolean result = false;
          if (myPlugType = "Type1") and (myRange > nearestTypeOne)
              result=true;
          if (myPlugType = "Type2") and (myRange > nearestTypeTwo)
              result=true;
          return result;
     }
}
```

Examiner Comments

Slightly less than half of the students attempted this question, but of those that did more than half did quite well. Some students were confused by the class design from Q8 and this came through in their answer to this question.

SECTION D

QUESTION 10

a) i. $\sim A \wedge (\sim A \vee B)$

ii.

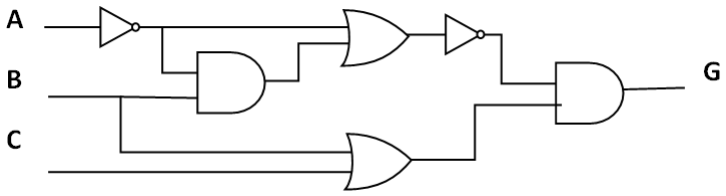
~ A V B
F
T
F
F

iii. A

b) i.

A	B	C	~B	~B ∧ C	A v B	F
0	0	0	I	0	0	0
0	0	I	I	I	0	I
0	I	0	0	0	I	I
0	I	I	0	0	I	I
I	0	0	I	0	I	I
I	0	I	I	I	I	I
I	I	0	0	0	I	I
I	I	I	0	0	I	I

ii.



iii. $H \equiv \sim((A \wedge \sim B) \vee \sim(B \wedge C))$

c)

Memory Address	Contents	Pseudocode	Explanation
01	0007	data	mem[01] contains value 7
02	0004	data	mem[01] contains 4
10	8A01	R[A] <- mem[01]	R[A] = 7
11	8B02	R[B] <- mem[02]	R[A] = 4
12	2CAB	R[C] ← R[A] - R[B]	R[C] = 7 – 4 = 3
13	0000	halt	

On program termination R[C] = **3**

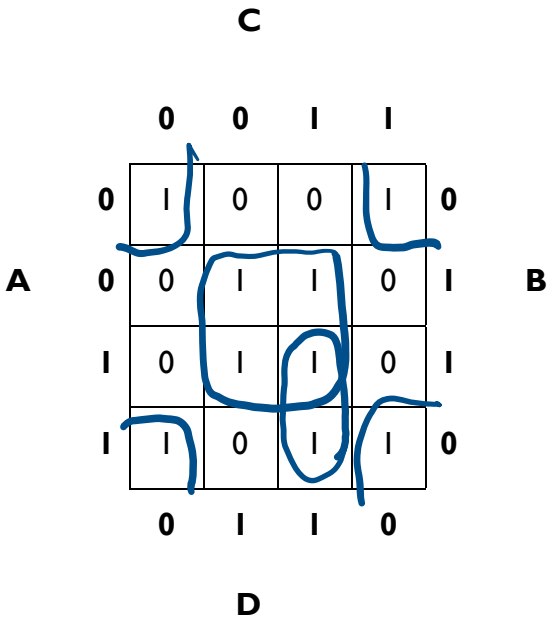
Examiner Comments

Question 10 was generally well answered by most students. The most challenging part was 10 b) ii. with only half the students obtaining full marks. A common mistake was to either position a NOT gate after the AND gate in the logic circuit or to neglect one of the NOT gates entirely.

The next most challenging question was 10 a) iii. with three-quarters of the students answering correctly.

QUESTION 11

a) i.



H ° (B ^ D) v (~B ^ ~D) v (A ^ C ^ D)

ii. ~A v (A ∧ B)

° (~A v A) ∧ (~A v B) L7
° T ∧ (~A v B) L14
° ~A v B L23

or ~A v (A ∧ B)
° ~A v (B ∧ A) L1
° ~A v B L27

b) In TOY, the opcode is identified by a single hex digit so there are only 16 opcodes. Branch negative isn't required because other opcodes can be used to obtain the same result. One approach would be to subtract from 0 and then use branch positive on the result. Another approach would be to use branch zero and branch positive in succession to skip the code that applies if the value is negative. For example:

```
10   8A01

11 CA14

12 DA14

13 *** R[A] must be negative ***

14       *** program continues ***
```

c) Java code is compiled into Java bytecode which can run by a JVM. The JVM converts the Java bytecode into a specific machine language to be executed on the host device

Examiner Comments

- 11 a) i. – Most students were able to earn at least partial marks for this question, but many either missed one of the clusters or made mistakes producing the logic expression. Although it didn't necessarily result in lost marks, very few students identified the corners as (~B ∧ ~D).
- 11 a) ii. – Answered reasonably well with half the students obtaining full marks.
- 11 b) – Not well answered. For full marks, students were expected to address both parts of this question.
- 11 c) – Not well answered. Many students simply rephrased the question. A number of students mentioned Java bytecodes but failed to explain what the JVM does with the bytecode.

QUESTION 12

a)

Memory Address	Contents	Pseudocode	Explanation
01	0003	data	Constant decimal 3
02	0000	data	Used for variable y
03	0003	data	Used for variable x
04	0001	data	Constant decimal 1
10	8A01	R[A] ← mem[01]	
11	8B02	R[B] ← mem[03]	variable x
12	2CBA	R[C] ← R[B] – R[A]	
13	DC16	if (R[C] > 0) pc ← 16	x > 3 ?
14	9B03	mem[03] ← R[B]	y = x
15	8D01		
16	5EBD	R[E] ← R[B] << R[C]	y = x * 2
17	9E03	mem[03] ← R[E]	
18	0000	halt	

b) Assuming the inputted values are in decimal, the value stored in memory will be

15_{10} $(0000\ 0000\ 0000\ 1111_2, 000A_{16})$

PC	Contents	R[A]	R[B]	R[C]	R[D]	R[E]	mem[01]
10	8AFF	11					
11	8BFF		12				
12	3DAB				8		
13	4EAB					7	
14	1CDE			15			
15	9C01						15

Explanation:

Opcode 3 is bitwise AND

R[A] 11 → 0000 0000 0000 1011

R[B] 12 → 0000 0000 0000 1100

R[A] AND R[B] 0000 0000 0000 1000 → 8

Opcode 3 is bitwise XOR

R[A] 11 → 0000 0000 0000 1011

R[B] 12 → 0000 0000 0000 1100

R[A] XOR R[B] 0000 0000 0000 0111 → 7

The purpose of this program is to perform a bitwise **OR** operation on two inputted values.

Assuming the inputted values are in hexadecimal, the value stored in memory will be

0013_{16} ($0000\ 0000\ 0001\ 0011_2$, 19_{10})

PC	Contents	R[A]	R[B]	R[C]	R[D]	R[E]	mem[01]
10	8AFF	0011					
11	8BFF		0012				
12	3DAB				0010		
13	4EAB					0003	
14	1CDE			0013			
15	9C01						0013

Examiner Comments

- 12 a) – This was answered reasonably well by most students that attempted it. For full marks, the resulting value for variable y had to be stored in mem[02].
- 12 b) – Very few students attempted this question and of those only a handful were able to complete the trace successfully and recognise that it was equivalent to the bitwise OR operator.

SECTION E

QUESTION 13

a) i. 00111100_2

ii. 39

iii. 70

b)

	1	**0**	1	**1**	0	
+	0	1	0	1	1	
1	**0**	0	**0**	0	1	

c)

i. **103 = 01100111_2**

ii. 96 n two's complement will be (reverse all bits) 10011111_2 + 1 = 10100000_2

iii.

$$01100111_2$$

$$+ \underline{10100000_2}$$

$$\mathbf{1}\ 00000111_2 \qquad \textbf{answer is 7}$$

d) **$0011\ 1100\ 1010\ 1011_2$**

e) The first place after the binary point is worth ½ then ¼ then 1/8 etc so ½ + 1/8 = 5/8 = 0.625

Examiner Comment

a) This question was completed correctly by nearly every student.

b) This question was attempted by nearly every student with it being correctly answered by 2/3 of the students but some made a simple mistake with the carry-on number when adding.

c) i was completed correctly by nearly every student. ii. and iii. were attempted by most students with 80% getting this one correct. There was some confusion with the 2's complement with a number of answers showing the 1's complement but not taking the next step to convert it to the 2's complement.

d) This question was answered correctly by the majority of the students. There was some confusion with the conversion from hex to binary and how it is displayed.

e) This question was answered correctly by the majority of the students. There was some confusion with the adding up of the binary points and conversions to fractions or decimals.

QUESTION 14

a) The largest number that can be stored in four-bit two's complement system is 7 ie 0111. Any calculation which results in a number larger than 7 will create an overflow ie force a 1 into the left most bit. e.g. 01002 + 01002 would result in 10002 As the first digit is a 1, this is a negative number.

b) 0.1011_2 = ½ + **1/8 + 1/16 = 11/16 = 0.6875**

c) Assuming 7 bits to store each ASCII character. Could store each of the four digits separately which would take 4bits x 4 = 16, however treating whole range from 0000-9999 would take 14 bits 2^{14} = 16384
So (14+14) 28 bits required for each plate. number of combinations is 26 x 26 x 10000 = 6760000.
Total bits = 6760000 x 28 = 189280000 bits.

d) $- 0.111 \times 2^{11}$

$= - 1792_{10}$

e) The conversion process yields 0.0100110 as shown below:

0.3 X 2 = **0**.6

0.6 X 2 = **1**.2

0.2 X 2 = **0**.4

0.4 X 2 = **0**.8

0.8 X 2 = **1**.6

0.6 X 2 = **1**.2 You can see from this point on the pattern will be repeated indefinitely

02 X 2 = **0**.4

Examiner Comment

a) This question was generally attempted by most students, however, most failed to illustrate the answer fully (as asked by the question). It appeared that either the question wasn't read fully or it wasn't understood what was being asked. Those who

answered the question were generally able to explain overflow error but more than half failed to explain it based on 4 digit 2's complement numbers as asked in the question.

b) This was generally completed correctly by most students.

c) This question confused all students with no-one getting full marks on this question. As there were 2 parts to the question, students seemed to either answer how much space it would take to store 1 number plate or how many different combinations of number plates there would be.

d) More than half of the students had difficulty answering this question. Only a few seemed comfortable with floating point numbers and conversion across to decimal.

e) This question was attempted by a large portion of students yet a common mistake that was made came down to answering the question in full. Quite a number of students would only explain 'why' and others would just convert 0.3, instead of explaining 'why' and using the conversion of 0.3 to explain their answer.

QUESTION 15

a) i. **per property**

Address 100 characters using 16-bit Unicode = 1600 bits

Cost 0 – 1000 = 10 bits

Occupancy 1 – 10 = 4 bits

Off street parking yes/no = 1 bit

_____

1615 bits

1000 properties @ 1615 bits per property = **1615000 bits**

ii. Cost 0 -1000.00 could store cost in cents and then would need 0 – 100000 which would need 17 bits

Would then require 1622 bits per property total **1622000 bits**

iii. Would not recommend storing as RAW as it would require too much storage and the quality would be overkill for a website. Would recommend converting to either JPG of PNEG as these will provide images of sufficient quality without consuming too much space.

b) i. int[ ] data = new int [7];

ii. Final value is 9.

01AE02 holds the size of the array which is 7.

The 7th mem location after 01AE02 is 1AE09 which contains value 9.

iii. char [ ] [ ] a = new char [2] [4];

Examiner Comment

a) i. About half the students attempted this question and only a small number achieved full marks. This question saw a mixed variety of responses with only a small group attempting every part of this question. There were a lot of errors being made that would have come from not reading the question correctly and understanding what it was asking (e.g., when storing the address of the property we need to find the space required for 1 character and then multiply it by 100 characters) and quite a number of students didn't multiply their final answer by 1000 to get the storage requirements for 1000 properties.

ii. Less than half the students attempted this question. This question seemed to confuse students and of those who answered, the majority didn't read the question fully. So, there were a lot of answers showing the minimum number of bits for the cost, but they also needed to have information for all the properties. The wording of this question may have confused some students.

iii. About half the students attempted this question and only a small number got full marks. As this question was asking first for your advice and then to explain your answer there was a number of students giving their advice but not explaining why they have come up with that answer.

b) i. Less than half the students attempted this question. Of those who attempted this question common mistakes were in how to declare a variable and then reading the above information to extract the correct value for the array variable data.

ii. Only a very small group of students attempted this question. Of the small group that answered the majority of them got the answer correct and they were able to explain their answer. There were also a number of students that incorrectly assumed 11. If the student was unable to answer the question b) i. then they had a hard time trying to answer this question.

iii. Less than half the students attempted this question. If they answered question b) i. they generally got this question as well.