ANSWERS

Section A

Question 1

(a) When a number is entered into “days” TextField
    set days to value in “days” TextField
    if days > 120
        set days to 120
        display “Maximum number of days is 120”

(b) When the “Calculate” button is pressed
    if size is ‘S’
        set rate to 55
    if size is ‘M’
        set rate to 70
    if size is ‘L’
        set rate to 90
    set cost to (rate x days)
    if gps is ‘Y’
        set gps_amount to (11 x days)
    else
        set gps_amount to zero
    set cost to (cost + gps_amount )
    display “Cost is” cost

Question 1 was well done. The only difficulty was students that didn’t read the question and only add $11 for the GPS rather than $11*days.

Question 2

When “Piece 1” button is pressed
    if random_pick is true
        set random_pick to false
        if (piece1 < piece2) and (piece1 + move > piece2)
            set move to (move + 1)
            set piece1 to (piece1 + move)
        if (piece1 equals piece2) and (not piece1 >= 50)
            set piece1 to zero
            set piece2 to zero
        display “Piece1 and Piece2 are ” piece1, piece2
if (piece1 >= 50) and (piece2 >= 50)
    display “Task completed in” count “ turns.”

Question 2 was also well done but no students identified a subtle situation at the end of the game. If “piece2” was off the board (ie piece2>50) then it was still possible for Piece1 == Piece2 but they were in fact not on the same square. The most common difficulty was not identifying that piece1 had to be less than piece2 prior to the addition of move (ie piece1 overtook piece 2). Many students just added 1 to piece1 if piece1>piece2.

Question 3

_Initially_
rides = 0
child = false
scary_rides = false
unlimited_rides = false

_When a number is entered into “Number of rides” TextField_
set rides to value in “Number of rides” TextField

_When “Child Ticket” button is pressed_
if child equals true
    set child to false
    display “Child ticket”
else
    set child to true
    set scary to false
    display “Not child ticket”

_When “Scary Rides” button is pressed_
if scary_rides equals true
    set scary_rides to false
    display “No scary rides”
else
    if child equals false
        set scary_rides to true
        display “Scary rides”

_When “Unlimited Rides” button is pressed_
if unlimited_rides equals true
    set unlimited_rides to false
    display “Not unlimited rides”
else
    set unlimited_rides to true
    display “Unlimited rides”

_When “Cost” button is pressed_
if rides <= 10
    set rate to 5
else
    if rides <= 20
        set rate to 4.5
    else
        set rate to 4.0
set cost to (rate * rides)
if unlimited_rides equals true
    set cost to 100
if scary_rides equals false
    set cost to (cost / 2)
if child is true
    set cost to (cost * 80 / 100)
display “Total cost of ticket is $” cost

<table>
<thead>
<tr>
<th>Number of</th>
<th>Cost:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Child Ticket</td>
<td>0</td>
</tr>
<tr>
<td>Scary Rides</td>
<td>Not a child</td>
</tr>
<tr>
<td>Unlimited</td>
<td>No scary rides</td>
</tr>
<tr>
<td>Not unlimited rides</td>
<td>Not unlimited rides</td>
</tr>
</tbody>
</table>

As expected, there was a broad range to the quality of answers in this section. Many students did not read the question carefully enough and made mistakes like:

- Not multiplying the number_of_rides*cost_per_ride.
- Not ensuring that children under 5 could not access the scary rides.
- Creating buttons for the three categories (1 to 11, 11 to 20, more than 20) rather than importing the number of rides and working out the cost per ride from this number.
- Forgetting about unlimited rides and the discounts that still applied.

In general I would advise students to:

- Take more care with planning and the initially-when set up
- Do all the calculations in the one “when calculate” rather than adjusting the calculation as each button/checkbox is activated. This is fraught with danger.
- Enable buttons to toggle (ie between adult and child and vice versa)
Section B

Question 4

(a)  
\[ a = 6 \]
\[ b = 6 + 6/3 = 6 + 2 = 8 \]
\[ c = \text{Math.max}(6, 8) = 8 \]

This question was handled pretty poorly. Most mistakes involved not using BODMAS as a basis to the question so that the 6+6 became 12 and then the division by three was done to incorrectly arrive at the answer 4. This had flow on effects to the next question making the max function choose between 4 and 6 rather than 6 and 8. If a student described the function of the max method they received some marks even if the final answer were 6 and not the expected 8.

(b)  
\[ p = 7 \text{ so } (p > 7) \text{ is false so switch statement uses } p = 7 \text{ this means } q = 200 \]

On the whole this question was pretty well answered. Some students did not believe the switch statement executed and gave incorrect answers and some did not realise that 7 is not greater than 7 so added the 3 to p. These answers were in the minority.

(c)  

<table>
<thead>
<tr>
<th>( n )</th>
<th>( \text{finished} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>false</td>
</tr>
<tr>
<td>5</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>true</td>
</tr>
</tbody>
</table>

Final value of \( n = 7 \)

This question was very well answered. Very few made errors and those that did generally continued the trace table after the loop would have actually ended.

Question 5

(a)  
\[ d = 3 / 2 = 1 \text{ (integer division)} \]
\[ e = 3 / 2.0 = 1.5 \]

This question was very troublesome for students. The first of the a questions was very rarely identified as 1 as many thought the end result being a double somehow promoted the two integers being divided before the division was completed. This was an incorrect assumption. Of the second equation most people got it correct though errors had it incorrectly resulting as a 1 rather than the expected 1.5 because of the division as a double.

(b)  

<table>
<thead>
<tr>
<th>( k )</th>
<th>( s )</th>
</tr>
</thead>
<tbody>
<tr>
<td>‘A’</td>
<td>“A”</td>
</tr>
<tr>
<td>‘B’</td>
<td>“BA”</td>
</tr>
<tr>
<td>‘C’</td>
<td>“CBA”</td>
</tr>
<tr>
<td>‘D’</td>
<td>“DCBA”</td>
</tr>
</tbody>
</table>
Final value of s is "DCBA"
This question was answered very poorly. Little attention was made to the detail of the question and there were a multitude of incorrect answers

1. Incorrectly ordering the answer e.g. ABCD
2. Adding +s to the string e.g. D+sC+sB+sA+s
3. Giving an answer as some numerical factor (based on the numerical value of the chars I suppose). The answer is a string, not a number.
4. Simply giving a one letter answer
5. Not breaking out of the loop when the char 'E' was reached e.g. EDCBA
6. Dropping the A from the sequence e.g. “DCB”

(c)

<table>
<thead>
<tr>
<th></th>
<th>x</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>50</td>
</tr>
<tr>
<td>1</td>
<td>100</td>
</tr>
<tr>
<td>2</td>
<td>150</td>
</tr>
<tr>
<td>3</td>
<td>200</td>
</tr>
</tbody>
</table>

Output of drawString method ... 300

This trace table was poorly done. The trace tables tended to show different answers to the drawn Applet and also the trace showed changes in numbers that would not have occurred at that point in the cycle. Many people also took that the j+1 mention in the array meant add one to the variable in array position j rather than select the value of position j+1 in the array.

Question 6

(a) The variables d is only used in the method move and it is used to store a temporary value. This means it is fine to declare it locally to move.
The variable direction is only used in the method move but it needs to hold its value from one call of the method to the next. It is used to assign d a value at the beginning of the method (char d = direction;) The only way direction can hold its value is to be a global variable.

For some reason very few people actually addressed the variable d in this question and focussed on direction. Most answers stated that the variable direction was declared globally so it could be used in other methods. The code provided was for a complete program and direction is only used in one method. Its use as a constant that needed to keep the previous state of the program was largely overlooked. Ignoring the d portion of this question meant half marks at best but those that did attempt talking about d got confused and very few came to the real reason why it was a local variable.
(b) The “worm” follows the instructions in the instruction array \{'L', 'F', 'L', 'R'\}. The worm starts with the head facing south and turns \textbf{Left (East)}, then \textbf{Forward (East)}, then \textbf{Left (North)}, then \textbf{Right (East)}. As the rest of the worm follows the head the final position is given below.

\[
\begin{array}{|c|c|c|}
\hline
# & # & # \\
# & # & # \\
\hline
\end{array}
\]

This question was unfortunately worded and very few students that attempted the question actually understood the meaning of it. The question was designed for the student to literally describe the process the program would take using the instruction set laid out in the \textbf{instructions} array. Due to some poor variable names this became convoluted and confusing. Students attempted a high level discussion of programming principles as opposed to an actual functional look at the question. Also, very few students produced a drawing of the Applet that was requested. Again this may have been due to poor formatting of the question as there was no fake Applet screen to draw this in as with other questions that asked similar questions and also the preceding portion of the question had seemed to confuse the student. There were very few pictures of the Applet done and even fewer that were correct. Many people that drew something tried to draw just the contents of the array (5 ‘#’ characters). Unsurprisingly this question was very poorly answered.

\textbf{Section C}

\textbf{Question 7}

a)

\begin{array}{|c|c|c|}
\hline
\textbf{aButto} & 1 & \textbf{bButto} & 0 & \textbf{none} \\
\hline
\end{array}

b)

\begin{array}{|c|c|c|}
\hline
\textbf{aButto} & 2 & \textbf{bButto} & 2 & \textbf{Lost!} \\
\hline
\end{array}

c)

\begin{array}{|c|c|c|}
\hline
\textbf{aButto} & 3 & \textbf{bButto} & 3 & \textbf{Won!} \\
\hline
\end{array}

Generally answered well.
Question 8

a) This Quest
This Question
This Question TROLLS!

Many students struggled with substring, particularly for the second version of string3

b)

This was very challenging to mark, as most students did not show working out and were not very good at scale drawings.
Many students did not show familiarity with drawOval.
A common mistake was using the x and Y as the centre point of the oval where they are actually the top left corner – see image provided.

Points were given when a relationship could be seen in the size of each of the circles, and roughly starting in the correct position.

Question 9

a) Catalogue  cat = new Catalogue();

b)  cat.addCategory(“Building”);
    cat.addCategory(“Office”);

        cat.addltem(“Building”, “Plumber”);
        cat.addltem(“Building”, “Carpenter”);
        cat.addltem(“Office”, “Clerk”);

c)  g.drawString(Clerk + “ is from “ + cat.findCategory(“Clerk”),50,50);

OR

String category = cat.findCategory(“Clerk”);
This was marked in two parts:
1 – using the method appropriately
2 – accepting/storing the result by using a drawstring, variable or other output
d) public boolean checkCategory(String Category)
{
    int i = 0;
    while (!store[i][0].equals(Category) && !store[i][0].equals(""))
        i=i+1;
    return store[i][0].equals(Category);
}

Common mistakes:
Comparing strings using == Strings must me compared using a method such as equals()
The method must return a Boolean value – this means writing public boolean... in the method definition, as well as returns ... at the end of the method body

Overall this question was done quite well.

Section D

Question 10

a) \( F \equiv (\sim B \land C) \lor (A \lor B) \)

\[
\begin{array}{cccc|c|c|c|c}
 A & B & C & \sim B & \sim B \land C & A \lor B & F \\ \hline
 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
 0 & 0 & 1 & 1 & 1 & 0 & 1 \\
 0 & 1 & 0 & 0 & 0 & 1 & 1 \\
 0 & 1 & 1 & 0 & 0 & 1 & 1 \\
 1 & 0 & 0 & 1 & 0 & 1 & 1 \\
 1 & 0 & 1 & 1 & 1 & 1 & 1 \\
 1 & 1 & 0 & 0 & 0 & 1 & 1 \\
 1 & 1 & 1 & 0 & 0 & 1 & 1 \\
\end{array}
\]

b) \( G \equiv (\sim A \lor (B \land \sim A)) \land (C \lor B) \)

\[
\text{Diagram}
\]

\( \sim Q \lor (P \land Q) \equiv \sim Q \lor P \quad \text{[L27 – Complement Rule]} \)

Or also accepted
Using LHS
\( (\sim Q \lor P) \land (\sim Q \lor Q) \quad \text{[L7 – Distribution Law]} \)
\( (\sim Q \lor P) \land (T) \quad \text{[L14 – Excluded Middle]} \)
\( \sim Q \lor P \equiv \text{RHS} \)
Most students had a go at this question. In general (a) and (b) were answered reasonably. Students who got 2 of these parts got a C or they had to provide supporting evidence in later questions. Many students could receive help in drawing circuit gates and using the logic laws.

**Question 11**

a) i) ii) 

\[ H = (\neg A \land \neg C) \lor (B \land D) \lor (\neg B \land C \land \neg D) \]

b) 

```java
int a = 3;
int b = 0;
if (a < 2)
    b = 4;
```

Question 11 (a) (i) was attempted by most students and overall was okay. Many students then had difficulty finding the simplest expression and so answers that treated the outside two 1s as individuals or multiple pairs were considered favourably. Many students failed to draw the circuit.

Part (b) the Toy question was very poorly answered with many students not even making an attempt. Of those that did many did not write Java code. Even amongst those that answered the question “correctly” had quite strange if statements (eg if \((a-2) \neq 0 \land (a-2) > 0\) \{ \} else \(b = 4;\). I feel that this is an area that teachers may need to make some effort with.

**Question 12**

a) The memory of the computer can store much larger amounts of data than the registers in the CPU but this means they are slower to access. The address of the memory location has to be sent to a register and then the data in the location retrieved to another register which then has to be transferred to another register to be used. All the registers in the CPU are wired in for very fast immediate access.

b) The numbers to be added would be loaded from memory into registers. The Add instruction would then be used to add the numbers. The result would then be stored in memory.

c) The two processes are basically the same both need to load the data into the CPU, carry out the addition and store the result in memory. The TOY instructions would be likely to take longer
because of the need to complete the full fetch/decode/execute cycle for each of the four instructions. The instruction to add directly to memory would only need one full fetch/decode/execute cycle to complete the instruction.

d) The dedicated load and store instructions can be made more efficient and so faster. Considering the last question if there were more operations to be completed on the data the time saved accessing the working data in the cache would be much quicker than the more complex instructions that work by using data in memory even considering the savings in clock cycles to perform the instructions of a CISC based system. The RISC based system would also be smaller allowing greater cache space on the CPU.

There were quite a few students who did not answer this question. For those that did part (a) was answered reasonably well and (b) quite well and this quite often enabled a student who had struggled with the toy part of the question to get their B. A good answer on this question definitely separated the A students. Most students did not answer part (c) well when asked to use the Fetch/Decode/Execute cycle to explain the difference.

Section E

Question 13

a) Fill in the four missing bits in the following binary addition.

```
  1  0  1  0  1
+  1  1  1  0  1

  1  1  0  0  1  0
```

Comment: A surprising number of students got this wrong.

b) i) Convert binary 101101 to hexadecimal = 5D

Comment: A large number of students converted to decimal rather than hexadecimal.

ii) If 101101 is stored in a 7 bit word using twos completed rep. what value.
Is negative because of leading one so take twos complement.
0100010 + 1 = 0100011 = 35 so number is -35 in decimal.

Comment: generally well answered. Some candidates just wrote 35, or gave the binary pattern for 35.

iii) If 01011101 is an ASCII character it will be ‘]’ (code 93).

Comment: Well answered.
c) Given the following table convert 0.72 to a binary number to 6 places = 0.101110

Comment: Straightforward, but many reversed the answer or gave more or less than six places.

Question 14

a) A floating point number splits the available bits into three parts. Sign, magnitude and exponent which allows a number to be represented as a floating point number eg. $0.1 \times 2^{01111}$. This format of number can easily create a number larger than 32767.

Comment: Answers showed a poor understanding of the impact of the exponent in controlling the range of numbers that can be represented.

b) $0111111 + 1 = 10000000$ so adding one to the largest value (causing integer overflow) gives the representation of a negative number in twos complement due to the leading one.

Comment: Generally well answered, although many candidates didn’t give an example as requested, or added two negative numbers to give a ninth bit.

c) Total characters would be $26 \times 2 = 52$. 52 in binary is 110100 so 6 bits are required.

Comment: Answers ranged from 5 bits to 4096 bits. Many candidates insisted that ASCII codes had to be used.

Question 15

a) i) $800 \times 600 \times 3 \times 8 = 11520000$ bits.

ii) $800 \times 600 \times 4 \times 7 = 1344000$ bits.

iii) $800 \times 600 \times 4 \times 8 = 15360000$ bits.

Comment: A large number of candidates just wrote a number (mostly incorrect), without showing how it was derived. This ensured they didn’t gain an A. It is also important to read the question carefully, which asked for the number of bits (not bytes).

Because the amount of data space saved (4 bits per pixel) would be insignificant compared to the improvement in processing time gained by having all values stored in bytes.

Comment: Few candidates commented on the additional processing required to unpack data that didn’t fall on byte boundaries.
(b) Comment: Most candidates were able to represent the array, although a number described it in terms of a memory map in TOY. A large number of candidates missed part (ii) of the question, or placed 4 in the first array (or in all 10 locations).
**Award Distribution**

<table>
<thead>
<tr>
<th></th>
<th>EA</th>
<th>HA</th>
<th>CA</th>
<th>SA</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>This year</td>
<td>14% (18)</td>
<td>15% (19)</td>
<td>39% (50)</td>
<td>31% (40)</td>
<td>127</td>
</tr>
<tr>
<td>Last year</td>
<td>15% (21)</td>
<td>20% (28)</td>
<td>24% (33)</td>
<td>41% (56)</td>
<td>138</td>
</tr>
<tr>
<td>Last year (all examined subjects)</td>
<td>11 %</td>
<td>19 %</td>
<td>39 %</td>
<td>31 %</td>
<td></td>
</tr>
<tr>
<td>Previous 5 years</td>
<td>10 %</td>
<td>18 %</td>
<td>33 %</td>
<td>39 %</td>
<td></td>
</tr>
<tr>
<td>Previous 5 years (all examined subjects)</td>
<td>11 %</td>
<td>19 %</td>
<td>39 %</td>
<td>30 %</td>
<td></td>
</tr>
</tbody>
</table>

**Student Distribution (SA or better)**

<table>
<thead>
<tr>
<th></th>
<th>Male</th>
<th>Female</th>
<th>Year 11</th>
<th>Year 12</th>
</tr>
</thead>
<tbody>
<tr>
<td>This year</td>
<td>94% (120)</td>
<td>6% (7)</td>
<td>31% (40)</td>
<td>69% (87)</td>
</tr>
<tr>
<td>Last year</td>
<td>96% (132)</td>
<td>4% (6)</td>
<td>46% (63)</td>
<td>54% (75)</td>
</tr>
<tr>
<td>Previous 5 years</td>
<td>94%</td>
<td>6%</td>
<td>47%</td>
<td>53%</td>
</tr>
</tbody>
</table>