Christ Church
Grammar School

Computer Science Department
Year 11 Computer Science

Year 11 ATAR Computer Science
In-Class Theory Test

Programming

Marking Guide

TOTAL ____ / 57

Question 1 (5 marks)

Given A is TRUE, B is FALSE and C is TRUE, evaluate the following Boolean expressions:

A AND B FALSE

A OR C TRUE

(A OR C) AND (A OR B) TRUE

NOT A AND C FALSE

(A AND B) OR (NOT A AND C) FALSE

Question 2 (3 marks)

Given A is 3, B is 4 and C is 5, evaluate the following expressions:

A == B FALSE

A != C TRUE

(A > C) OR (B < C) TRUE

Question 3 (4 marks)

Give an example of each of the following data types:

integer	12
string	word
Boolean	TRUE
float	5.3

Question 4 (5 marks)

Consider the following algorithms and identify the **final** output from each algorithm.

```
num = 6
IF num > 0
    PRINT(num * 2)
ELSE IF num < 0:
    PRINT(num / 2)
ELSE
    PRINT(num)
END IF
```

Output: 12

```
n = 4
count = 1
sum = 0
WHILE count <= n
    sum = sum + (count * count)
    count += 1
END WHILE
PRINT(sum)
```

Output: 30

```
x = 5
y = 10
z = 0
IF x > y
    z = x - y
ELSE
    z = x + y
END IF
PRINT(z)
```

Output: 15

```
a = 10
b = 20
c = 30
d = 0
IF a > b AND a > c THEN
    d = a
ELSE IF b > a and b > c THEN
    d = b
ELSE
    d = c
END IF
PRINT(d)
```

Output: 30

```
a = “one”
b = “two”
c = “three”
c = a
b = a
a = c
PRINT(a, b, c)
```

Output: one one one

Question 5 (4 marks)

Vikki is always looking to save money for her business by pirating software and has decided to use a pirated copy of a popular accounting package to keep track of her business’s finances. Discuss two ethical considerations with her behaviour.

For each consideration:

2 marks complete discussion of ethical consideration
1 mark superficial comment about consideration
NOTE: Needs to be ethical consideration, not legal

Ethical considerations include:

- Proper recognition of developer’s efforts: developers of commercial software create new programs for economic gain and this needs to be recognised in order for them to be properly rewarded. Pirating software means that developers do not receive the proper economic rewards so are unable to create more software
- Security concerns: using pirated software may open her company up to installing malware which could cause her company financial loss
- Technical support: if Vikki uses pirated software and has trouble using the software for her accounts or has an issue with her accounts she will not be able to access technical support which may cause her business issues.

Question 6 (9 marks)

(a) Identify and describe three types of coding errors that occur when programming. (6 marks)

For each type of error:

1 mark each error identified
2 marks description of the error
1 mark superficial comment about error

Syntax Errors

Syntax errors (also known as compile-time errors) are caused by program statements that do not conform to the rules of the language. Some common syntax errors include:

- Missing or incorrect punctuation (e.g. not closing brackets, missing commas or semicolons etc)
- Typographical errors for reserved words (e.g. typing PRNT instead of PRINT)
- Incomplete program statements (e.g. leaving out the condition statement in an IF statement)

Syntax errors are detected during the compilation/interpretation of a program.

Run-Time Errors

A run-time error is an error that occurs during the execution of a program that prevents the program from running correctly. One common type of run-time error occurs when the program attempts to perform a calculation for which there is no answer (e.g. dividing by zero). Other run-time errors may be due to how numbers are stored by the computer and limitations on the memory allocated to the numbers. For example, an integer addition that will produce a result outside the expected range of an integer (often -32768 to 32767) will cause an error.

Logic Errors

A logic error is an error the that occurs when the program executes however produces unexpected output. Logic errors are often the most difficult to debug and software requires substantial testing to ensure there are no logic errors.

(b) Identify which type of error is usually the most difficult to find and fix. Justify your choice. (3 marks)

1 mark error type identified
2 marks full justification
1 mark superficial comment

Error Type: Logic error

A logic error is generally more difficult to find because the program runs without causing so does not give a clear indication as to where the error might be. Syntax errors and run-time errors both give an error message which provides some information about the error, making it easier to track down and fix.

Question 7 (6 marks)

Ellyse loves words, and the longer the better! Help her find the longest word in her book by writing an algorithm.

Your algorithm should:

- Read in a list of words. The list could be of any length and the program should stop asking for a new word when the user enters an empty string
- Find the longest word
- Print the longest word

```
MODULE LongestWord

    longest_word = ‘’
    new_word = ‘’

    INPUT(new_word)
    WHILE new_word != ‘’
        IF words[i].length > longest_word.length
            longest_word = words[i]
        END IF
    END WHILE

    PRINT(longest_word)

END LongestWord
```

1 mark initialise variables
1 mark reading words
1 mark using WHILE loop
1 mark keeping track of longest word
1 mark checking length of current word with longest word
1 mark printing longest word

Question 8 (4 marks)

Jake is writing a program to calculate the amount of interest that he needs to pay on his home loan. Discuss two reasons why it would be useful for him to use a constant called *interest_rate* in his program.

For each reason:

2 marks detailed discussion with reference to question
1 mark superficial comment about reason

Reasons could include:

- Readability of code: using constants can make the code easier to read as Jake can use a meaningful identifier in his program when he wants to refer to the interest rate, rather than hard coding a number which may have a different meaning in different contexts.
- Maintainability of code: using a constant for the interest rate means that in future if the interest rate changes he only needs to make a change to one line of code, rather than changing each occurrence of the interest rate throughout the entire program.

Question 9 (6 marks)

Discuss three benefits of using modularisation when writing software.

2 marks each benefit discussed
1 mark superficial comment about a benefit

Benefits include:

- By creating modules for tasks that are completed multiple times, it allows that code to be easily be reused and reduces the need to repeat the same lines of code multiple times
- it allows more people to work on a project. Since the project is broken down into several parts, each person can work on separate modules
- by breaking a large complex problem down into smaller problems it make it easier to solve the overall problem, as generally smaller problems are easier to solve than one large problem
- it makes it easier to read algorithms and programs as using appropriate names for modules should describe what each module does, rather than having to interpret the code
- as each module performs a single function it makes it quicker and easier to identify the module that is causing a problem and fix the error

Question 10 (4 marks)

Jake loves playing with numbers and wants some programs to help him learn more about them.

The first game he calls *Evens*. This game involves getting a list of numbers and finding the sum of all the even numbers in the list.

For example, given the list [3, 6, 4, 1, 2], the sum would be 12. Given the list [1, 2, 3, 4, 5, 6, 7, 8], the sum would be 20.

Complete the function *Evens* below that takes in a list of numbers, *numbers*, as a parameter and returns the sum of all the even numbers in the list.

```
FUNCTION Evens(numbers)
    sum = 0

    FOR i = 0 TO numbers.length – 1
        IF numbers[i] % 2 == 0
            sum = sum + numbers[i]
        END IF
    END FOR

    RETURN sum
END EvenSum
```

1 mark initialise sum
1 mark loop through list correctly
1 mark check for even number
1 mark add number to sum

Question 11 (7 marks)

Mr Farmer want so make sure students don't get too sunburnt during PE lessons but is having trouble reading the UV sensor on the gym wall from his office!

Write an algorithm called *UV_rating* that will ask the user to enter an integer that will represent the current UV level, and print out the UV rating based on the table below.

UV Level	Rating
1-2	low
3-5	moderate
6-7	high
8-10	very high
11 and above	extreme

```
MODULE UV_Rating
    level = 0
    rating = ‘’

    INPUT(level)

    IF level <= 2
        rating = “low”
    ELSE IF level <= 5
        rating = “moderate”
    ELSE IF level <= 7
        rating = “high”
    ELSE IF level <= 10
        rating = “very high”
    ELSE
        rating = “extreme”

    PRINT(rating)
END UV_Rating
```

1 mark start and end module appropriately
1 mark initialise variables
1 mark input level
3 marks correct multi-way selection and rating
1 mark print output

End of questions