|  |  |  |  |
| --- | --- | --- | --- |
| **Worksheet 1** | **2.1** | **Computer Architecture** | **Basic Hardware Components** |

1. a) memory

b) processor

c) port for input / output

d) data and address buses

e) external drive

1. ALU processes data by performing basic mathematical and logical operations such as addition, subtraction, multiplication and division.

The control unit follows instructions and decides when data should be stored, received or transmitted by different parts of the computer.

1. a) bidirectional

b) unidirectional

1. from the computer's processor to the computer's memory.
2. 4 × 8 = 32 bits
3. a) 2,017 × 1,024 = 2,065,408 bytes

b) 9 × 1,024 × 1,024 × 1,024 = 9,663,676,416 bytes

c) 65 × 1,024 × 1,024 = 68,157,440 bytes

d) 2,020 x 1,000 = 2,020,000 bytes

1. a) 3,072 / (1,024) = 3 MiB

b) 5,242,880/ (1,024 × 1,024) = 5 MiB

c) 16,777,216/ (8 × 1,024 x 1,024) = 2 MiB

d) 7,864,320 / (1,024 x 1,024) = 7.5 MiB

|  |  |  |
| --- | --- | --- |
| **Device** | **Input Device** | **Output Device** |
| Web camera | ✓ |  |
| Earphones |  | ✓ |
| Keyboard | ✓ |  |
| Microphone | ✓ |  |
| Mouse | ✓ |  |
| Printer |  | ✓ |
| Scanner | ✓ |  |
| Touch screen | ✓ | ✓ |

1. RAM: volatile

ROM: non-volatile

Digital versatile disc (DVD): non-volatile

Hard disk: non-volatile

Secure Digital (SD) cards: non-volatile

1. Long-term archival of more than 20 TiB of data: Magnetic Tape

Fast internal storage for a laptop: Solid State Drive

Portable way to transfer less than 32 GiB of data from one device to another: USB flash drive

|  |  |  |
| --- | --- | --- |
| **Usage** | **Choice of Computer** | **Reasons** |
| photo editing | Computer 2 | As the computer will need to store and load large graphical files, a relatively large amount of storage (512GB) and memory (8GB) is needed.  As photos will be regularly transferred from a camera to the computer by memory card, the computer will need a media card reader. |
| home use | Computer 3 | As document processing and simple web browsing are not processor-intensive operations, a relatively slow processor (1.6 GHz) will be sufficient.  As office documents do not usually take up a lot of space, a relatively small amount of storage (32GB) will be sufficient. |
| gaming | Computer 1 | As games can take up a lot of space, a relatively large amount of storage (1TB hard disk) is needed. Some games perform poorly if loading times are too slow, so the computer should ideally have a solid-state drive (512GB) as well.  As games are processor and graphics-intensive, the computer will need a fast processor (up to 4.9GHz) and dedicated graphics. |

|  |  |  |  |
| --- | --- | --- | --- |
| **Worksheet 2** | **3.1** | **Problem Analysis** | **Problem Statements** |

1. C

2. C

3. A

4. *Weights:* table of crates and their corresponding weights; each weight must be > 0.

*Capacity:* maximum weight that a net can carry; must be > 0

5. *Letters1:* sequence of English letters; must not be empty

*Letters2*: sequence of English letters; must not be empty

6. *Sequences*: list of at least two English letter sequences; each letter sequence must not be empty

7. A

8a. *Class list*: list of all register numbers for students in your class; must not be empty; each register number must be a positive whole number

*Football list*: list of register numbers for students who signed up for football; may be empty; each register number must also appear in *Class list*

*Basketball list*: list of register numbers for students who signed up for basketball; may be empty; each register number must also appear in *Class list*

8b. List of register numbers for all students in your class who have not signed up for either football or basketball **OR** list of register numbers that appear in *Class list* but do not appear in *Football list* or *Basketball list*

9a. - Empty list. Reason: there are no words in the list at all.

- A list with words of the same length, e.g. basket, square. Reason: all words in the list have the same length so none of them are the shortest.

9b. - The shortest word from word list (normal output).

- Output returns nothing as word list is empty.

- Output returns nothing as all words in the word list have the same length.

|  |  |  |  |
| --- | --- | --- | --- |
| **Worksheet 3** | **3.1** | **Problem Analysis** | **Modularity and Generalisation** |

1. GOTHIC

2. B

3. B

4.

|  |  |  |
| --- | --- | --- |
| **Sub-Problem** | **Input** | **Output** |
| #1 | * *First gift*: first gift in *price list* and its corresponding price | * *Result #1*: Most expensive gift from first item in price list |
| #2 | * *Result #1*: Most expensive gift from first item in price list * *Second gift*: second gift in *price list* and its corresponding price | * *Result #2*: Most expensive gift from first two items in price list |
| #3 | * *Result #2*: Most expensive gift from first two items in price list * *Third gift*: third gift in *price list* and its corresponding price | * *Result #3*: Most expensive gift from first three items in price list |
|  | *…and so on…* | *…and so on…* |

5. D

6. Step 1: If *number* is less than *divisor*, proceed to Step 3. Otherwise, proceed to Step 2.

Step 2: Subtract *divisor* from *number* and proceed to Step 1.

Step 3: Output *number* as the final answer.

|  |  |  |  |
| --- | --- | --- | --- |
| **Worksheet 4** | **3.2** | **Algorithm Design** | **Flowcharts** |

1.

|  |  |  |
| --- | --- | --- |
| **Name** | **Symbol** | **Brief Description** |
| Terminator |  | It represents the beginning or end of a sequence of steps, and usually contains either the START or STOP command. |
| Data |  | It represents the step of either receiving input data from outside the algorithm using the INPUT command or producing output from within the algorithm using the OUTPUT command. |
| Decision |  | It represents a step involving a question. The outgoing arrows represent the possible outcomes to the question and are usually labelled “Yes” and “No”. There may be two or three outgoing arrows depending on the number of possible outcomes. Only one of these outgoing arrows should be followed when performing the algorithm. |
| Process |  | It represents a step involving an action or operation. This usually involves changing the value of a variable or performing more complex actions. |

2a. True

2b. False

2c. False

3.

|  |  |
| --- | --- |
|  | Sequence Construct |
|  | Selection Construct |
|  | Iteration Construct |

4a.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| count | number | total | x | average | OUTPUT |
| 1 |  |  |  |  |  |
|  |  | 0 |  |  |  |
|  |  |  | 0 |  |  |
|  | 16 |  |  |  |  |
|  |  | 16 |  |  |  |
|  |  |  | 1 |  |  |
| 2 |  |  |  |  |  |
|  | -2 |  |  |  |  |
| 3 |  |  |  |  |  |
|  | 0 |  |  |  |  |
| 4 |  |  |  |  |  |
|  | 7 |  |  |  |  |
|  |  | 23 |  |  |  |
|  |  |  | 2 |  |  |
| 5 |  |  |  |  |  |
|  | 1 |  |  |  |  |
|  |  | 24 |  |  |  |
|  |  |  | 3 |  |  |
| 6 |  |  |  |  |  |
|  |  |  |  | 8 |  |
|  |  |  |  |  | 8 |

4b. Given 5 numbers, the algorithm will output the average of the positive numbers only.

5.

|  |  |  |  |
| --- | --- | --- | --- |
| **count** | **number** | **sum** | **OUTPUT** |
| 0 |  |  |  |
|  | 0 |  |  |
|  |  | 0 |  |
|  | 16 |  |  |
|  |  | 16 |  |
| 1 |  |  |  |
|  | 7 |  |  |
| 2 |  |  |  |
|  | -3 |  |  |
| 3 |  |  |  |
|  | 2 |  |  |
|  |  | 18 |  |
| 4 |  |  |  |
|  | 1 |  |  |
| 5 |  |  |  |
|  |  |  | 18 |

6.



|  |  |  |  |
| --- | --- | --- | --- |
| **Worksheet 5** | **3.2** | **Algorithm Design** | **Pseudo-code** |

1. a) i) 400

ii) 615

b)

|  |
| --- |
| INPUT hours\_worked hourly\_rate = 10 IF hours\_worked <= 40 THEN   weekly\_pay = hours\_worked \* hourly\_rate ELSE  normal\_pay = 40 \* hourly\_rate  overtime\_pay = (hours\_worked – 40) \* hourly\_rate \* 2  weekly\_pay = normal\_pay + overtime\_pay ENDIF OUTPUT weekly\_pay |

2.

|  |
| --- |
| INPUT age IF age < 18 THEN  OUTPUT "You are not eligible to drive" ELSE  OUTPUT "You are eligible to drive" ENDIF |

3.

|  |
| --- |
| INPUT grade  IF grade == "A1" THEN  OUTPUT "75% and above" ELSEIF grade == "A2" THEN  OUTPUT "70% to 74%" ELSEIF grade == "B3" THEN  OUTPUT "65% to 69%" ELSEIF grade == "B4" THEN  OUTPUT "60% to 64%" ELSEIF grade == "C5" THEN  OUTPUT "50% to 59%" ELSE  OUTPUT "Below 50%" ENDIF |

4. a)

|  |  |  |  |
| --- | --- | --- | --- |
| **Count** | **Sum** | **Number** | **OUTPUT** |
| 0 |  |  |  |
|  | 0 |  |  |
|  |  | 6 |  |
|  | 6 |  |  |
| 1 |  |  |  |
|  |  | 7 |  |
| 2 |  |  |  |
|  |  | 8 |  |
| 3 |  |  |  |
|  |  | 9 |  |
|  | 15 |  |  |
| 4 |  |  |  |
|  |  | 10 |  |
| 5 |  |  |  |
|  |  |  | 15 |

b) The algorithm accepts five numbers as input and outputs the sum of only the numbers that are divisible by 3.

5.

|  |
| --- |
| count = 0  total\_cost = 0  INPUT membership  INPUT number  WHILE count < number  INPUT item\_cost  total\_cost = total\_cost + item\_cost  count = count + 1  ENDWHILE  IF membership == "Y" THEN  discount = 0.15 \* total\_cost  ELSE  discount = 0  ENDIF  final\_cost = total\_cost – discount  OUTPUT total\_cost, discount, final\_cost |

6.

|  |  |  |  |
| --- | --- | --- | --- |
| **count** | **total** | **mark** | **OUTPUT** |
|  | 0 |  |  |
| 1 |  |  |  |
|  |  | 10 |  |
|  | 10 |  |  |
| 2 |  |  |  |
|  |  | 20 |  |
|  | 30 |  |  |
| 3 |  |  |  |
|  |  | 30 |  |
|  | 60 |  |  |
| 4 |  |  |  |
|  |  | 40 |  |
|  | 100 |  |  |
| 5 |  |  |  |
|  |  | 50 |  |
|  | 150 |  |  |
|  |  |  | 150 |

7.

|  |
| --- |
| total = 0  highest = -1 lowest = 51 FOR count = 1 to 5  INPUT mark  IF mark > highest THEN  highest = mark  ENDIF  IF mark < lowest THEN  lowest = mark  ENDIF  total = total + mark NEXT count average = total / 5 OUTPUT total, average, highest, lowest |

**Note:** In general, it is not a good practice to use the counter variable (named "count" in this case) for a FOR-loop outside the loop.

8.

|  |
| --- |
| INPUT N FOR count = 1 to N  IF count MOD 5 == 0 THEN  OUTPUT count  ENDIF NEXT count |

9.

|  |
| --- |
| INPUT N result = 0 FOR count = 1 to N  result = result + count \* count NEXT count OUTPUT result |

**Reflection questions**

* Give a situation where FOR…NEXT loop cannot be used.

When the number of iterations is not known and cannot be calculated before the loop starts.

|  |  |  |  |
| --- | --- | --- | --- |
| **Worksheet 6** | **3.2** | **Algorithm Design** | **Dry Runs, Trace Tables and Logic Errors** |

1. The algorithm is for counting the number of prime numbers from an input of ten integers.

2. Error 1: Line 1

Correction: Total = 0

Error 2: Line 2  
Correction: FOR Count = 0 to 4 **OR** FOR Count = 1 to 5

**Note:** For revised syllabus from 2021 Sec 3, the index for lists such as Number is taken to start from zero, so FOR Count = 1 to 5 is discouraged. However, for summative assessment, both options should be marked as correct as long as the logic presented is understood and correctly solves the problem.

Error 3: Line 3  
Correction: Total = Total + Number[Count]

Error 4: Line 4  
Correction: Line 4 not necessary. Remove statement.

Error 5: Line 6  
Correction: Average = Total / 5

3.

|  |  |  |  |
| --- | --- | --- | --- |
| **counter** | **largest** | **x** | **OUTPUT** |
|  | 0 |  |  |
| 0 |  |  |  |
|  |  | 13 |  |
| 1 |  |  |  |
|  |  | 12 |  |
| 2 |  |  |  |
|  |  | 11 |  |
| 3 |  |  |  |
|  |  | 15 |  |
| 4 |  |  |  |
|  |  | 14 |  |
| 5 |  |  |  |
|  |  |  | 0 |

To correct the flowchart, the text ‘Is x < largest’ in the decision symbol should be changed to ‘Is x > largest’.

4.

START

STOP

smallest = 101

INPUT x

Is counter < 5

smallest = x

count = count + 1

Is x < smallest?

Yes

No

No

Yes

counter = 0

OUTPUT

smallest

|  |  |  |  |
| --- | --- | --- | --- |
| **Worksheet 7** | **4.1** | **Program Development** | **Python Syntax and Variables** |

1. **Hello, World**
2. **Hello, <name>**

**Note:** The name you entered previously will appear in place of <name>.

1. 4 spaces per level of indentation
2. An ‘unexpected indent’ message appears because there is unnecessary single space indentation on lines 2 and 3.
3. Refer to 7q6.py.

Corrected code:

|  |
| --- |
| **a = 5**  **b = 6**  **print(a,b)** |

1. The line **print("Line 4")** is indented incorrectly.
2. Refer to 7q8.py.

Corrected code:

|  |
| --- |
| **if True:**  **print("Line 1")**  **print("Line 2")**  **print("Line 3")**  **print("Line 4")**  **print("Line 5")** |

1. Refer to 7q9.py.

Corrected code:

|  |
| --- |
| **if True:**  **print("True")**  **else:**  **print("False")** |

10 (a)

1. 4 spaces
2. 12 spaces
3. 8 spaces

(b) The program checks and prints a message based on the BMI value. In this case, the BMI value provided on line 1 is 22.0, which is considered a healthy weight.

1. Actual Output:

|  |
| --- |
| **This is not a comment.** |

1. Actual Output:

|  |
| --- |
| **This is not a comment.**  **But the comments are finished now.** |

1. Rules for Python Identifiers

* Begin with a:
  + lowercase character '**a**' – '**z**' or
  + uppercase character '**A**' - '**Z**' or
  + **\_** and
* Followed by **alphanumeric** characters or '**\_**'
* Do not use **reserved** words
* Python is **case-sensitive**

1. Which of these variable names are allowed in Python? Put a tick for names that are allowed and a cross for names that are not allowed.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| box | √ |  | postal code | X |  | pi | √ |
| Box | √ |  | postal\_code | √ |  | PI | √ |
| BOX | √ |  | postal.code | X |  | alex | √ |
| Box2 | √ |  | postalCode | √ |  | Alex | √ |
| 1Box | X |  | postal# | X |  | 3Alex | X |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| GreatBigVariable | √ |  | 6x | X |
| greatBigVariable | √ |  | x6x | √ |
| great\_big\_variable | √ |  | all% | X |
| great.big.variable | X |  | #left | X |

1. Actual Output:

|  |
| --- |
| **9** |

19.

|  |  |
| --- | --- |
| **Code** | **Actual Output** |
| >>> **x = 8**  >>> **print(x)** | 8 |
| >>> **x = x + 1**  >>> **print(x)** | 9 |
| >>> **x = x + x**  >>> **print(x)** | 18 |

20. Likely names for constants

|  |  |
| --- | --- |
| Area | X |
| AREA | √ |
| Sum\_of\_triangles | X |
| SUM\_OF\_TRIANGLES | √ |

21. The print statement is before the assignment statement, which is incorrect.

22. There is a missing \* operator on line 3.

|  |  |  |  |
| --- | --- | --- | --- |
| **Worksheet 8** | **4.1** | **Program Development** | **Arithmetic and Relational Operators** |

1.

|  |  |  |
| --- | --- | --- |
| **Arithmetic Operation** | **Python Code** | **Actual Output** |
| Add 155 to 456 | 155 + 456 | 611 |
| Subtract 87 from 123 | 123 - 87 | 36 |
| Divide 256 by 8 | 256 / 8 | 32.0  **Note:** Python always returns a float when performing normal division |
| Multiply 47 by 13 | 47 \* 13 | 611 |
| Divide 63 by 21, then add 6 | (63 / 21) + 6 | 9.0 |

2.

|  |  |  |  |
| --- | --- | --- | --- |
| **Arithmetic Operation** | **Actual Output** | **Name of operator** | **Description** |
| 47 **//** 5 | 9 | floor division | Returns quotient when left-hand operand is divided by right-hand operand, rounded down to an integer |
| 24 **%** 5 | 4 | modulus | Returns remainder when left-hand operand is divided by right-hand operand |
| 2 **\*\*** 4 | 16 | exponent | Returns the left-hand operand raised to the power of the right-hand operand |

3.

|  |  |  |
| --- | --- | --- |
| **No** | **Operation** | **Actual Output** |
| 1 | a + b | 22 |
| 2 | a - b | 8 |
| 3 | a / b | 2.142857142857143 |
| 4 | a \* b | 105 |
| 5 | c // d | 2 |
| 6 | c % d | 1 |
| 7 | c \*\* d | 25 |

4.

|  |  |
| --- | --- |
| **Code** | **Actual Output** |
| >>> **a = 9**  >>> **b = 3**  >>> **a -= b**  >>> **print(a)** | 6 |
| >>> **a = 9**  >>> **b = 3**  >>> **a /= b**  >>> **print(a)** | 3.0 |
| >>> **a = 3**  >>> **b = 4**  >>> **a \*\*= b**  >>> **print(a)** | 81 |
| >>> **a = 4**  >>> **b = 3**  >>> **a %= b**  >>> **print(a)** | 1 |

5. Refer to 8q5.py. A possible solution is:

|  |
| --- |
| **# This program calculates the area of a triangle**  **base = float(input("Enter base of triangle: "))**  **height = float(input("Enter height of triangle: "))**  **area\_of\_triangle = 1/2 \* base \* height**  **print("The area of the triangle is: " + str(area\_of\_triangle))** |

6. Refer to 8q6.py. A possible solution is:

|  |
| --- |
| **# This program always returns an output of 3**  **number = int(input("Enter an integer from 1 to 10 (inclusive): "))**  **number = (number \* 2 + 6)/2 - number**  **print("The result is: " + str(number))** |

7. Refer to 8q7.py.

The second operation correctly computes the average as the addition operations within parentheses are performed before the division operation. For the first operation, the division operation is performed first.

8. Actual Output:

|  |
| --- |
| **math.sqrt(9): 3.0**  **math.sqrt(16): 4.0** |

1. The Math symbol for math.sqrt() is
2. Refer to 8q10.py. A possible solution is:

|  |
| --- |
| **# This program calculates the hypotenuse of a right-angled triangle**  **import math**  **base = float(input("Enter base of right-angled triangle: "))**  **height = float(input("Enter height of right-angled triangle: "))**  **hypo = math.sqrt((base\*\*2) + (height\*\*2))**  **print("The hypotenuse of the triangle is: " + str(hypo))** |

|  |  |
| --- | --- |
| **Boolean Statement** | **Actual Output** |
| **False or False** | **False** |
| **False or True** | **True** |
| **TRUE or True** | **Error** |
| **True or True** | **True** |

12.

|  |  |
| --- | --- |
| **Boolean Statement** | **Actual Output** |
| **Not True** | **Error** |
| **not False** | **True** |
| **NOT False** | **Error** |

13.

|  |  |
| --- | --- |
| **Boolean Statement** | **Actual Output** |
| **2 == 2 and 3 == 2** | **False** |
| **"hello" == "hello"** | **True** |
| **2 == 2 or 3 != 2** | **True** |
| **True and 1 == 1** | **True** |
| **False and 0 != 0** | **False** |
| **True or 1 == 1** | **True** |
| **"test" == "testing"** | **False** |
| **1 != 0 and 2 == 1** | **False** |
| **"test" != "testing"** | **True** |
| **"test" == 1** | **False** |
| **not (True and False)** | **True** |
| **not (1 == 1 and 0 != 1)** | **False** |
| **not (10 == 1 or 1000 == 1000)** | **False** |
| **not (1 != 10 or 3 == 4)** | **False** |
| **not ("testing" == "testing" and "Bala" == "Cool")** | **True** |
| **1 == 1 and (not ("testing" == 1 or 1 == 0))** | **True** |
| **"good" == "not bad" and (not (3 == 4 or 3 == 3))** | **False** |
| **5 == 5 and (not ("testing" == "testing" or "Python" == "Fun"))** | **False** |

1. Refer to 8q14.py

|  |
| --- |
| **x = int(input("Enter an integer between 1 and 10, inclusive: "))**  **if x >= 5:**  **print("Equal to or more than 5")**  **else:**  **print("Less than 5")** |

|  |  |  |  |
| --- | --- | --- | --- |
| **Worksheet 9** | **4.1** | **Program Development** | **Numeric Data Types** |

1. int, float

|  |  |  |
| --- | --- | --- |
| **No** | **Example** | **Numeric Type (int or float)** |
| a) | 10 | int |
| b) | 0.02 | float |
| c) | 99 | int |
| d) | 15.30 | float |
| e) | -786 | int |
| f) | -21.9 | float |
| g) | 2e15 | float |
| h) | -80. | float |
| i) | 33E-33 | float |
| j) | 435600 | int |

1. Actual Output of a + b:

|  |
| --- |
| **11.0** |

1. float
2. **4**
3. **8.0**
4. **3.142857142857143**
5. **3**
6. The code can just be pi = 3.14159. Since 3.14159 is already of type float, there is no need to perform any type conversion using float().

|  |  |  |  |
| --- | --- | --- | --- |
| **Worksheet 10** | **4.1** | **Program Development** | **Strings and Console Input/Output** |

1.

|  |  |
| --- | --- |
| **Method** | **Actual Output** |
| >>> **'Hello World'.endswith('ld')** | **True** |
| >>> **'2017'.startswith('20')** | **True** |
| >>> **'CS2017'.find('20')** | **2** |
| >>> **'CS2017'.find('21')** | **-1** |
| >>> **'PIZza'.lower()** | **'pizza'** |
| >>> **a = 'Hello'**  >>> **a.upper()** | **'HELLO'** |
| >>> **'PIZza'.islower()** | **False** |
| >>> **'CS2017'.isupper()** | **True** |
| >>> **' '.isspace()** | **True** |
| >>> **' Computing'.isspace()** | **False** |
| >>> **'Computing 2017'.isalnum()** | **False** |
| >>> **'Computing2017'.isalnum()** | **True** |
| >>> **'computing'.isalpha()** | **True** |
| >>> **'CS2017'.isalpha()** | **False** |
| >>> **'2017'.isdigit()** | **True** |
| >>> **'2017 '.isdigit()** | **False** |

2.

|  |  |
| --- | --- |
| **Operation** | **Actual Output** |
| >>> **'Hello' + "World"** | **'HelloWorld'** |
| >>> **x + y** | **'HelloWorld'** |
| >>> **x \* 2** | **'HelloHello'** |
| >>> **2 \* x** | **'HelloHello'** |
| >>> **x + " " + y** | **'Hello World'** |

**" "** represents a string containing a space.

1. Refer 10q3.py

|  |
| --- |
| **print(p + q\*2)** |

4.

|  |  |  |
| --- | --- | --- |
| **Code** | **Actual Output** | **What it does** |
| >>> **"man" in "maneater"** | **True** | Checks if a substring is in a string. |
| >>> **"eat" not in "theatre"** | **False** | Checks if a substring is not in a string. |
| >>> **len("maneater")** | **8** | Returns the length of a string. |

5.

|  |  |
| --- | --- |
| **Code** | **Actual Output** |
| s[0] | **'C'** |
| s[3] | **'p'** |
| s[-2] | **'n'** |
| s[0:2] | **'Co'** |
| s[1:2] | **'o'** |
| s[:2] | **'Co'** |
| s[::2] | **'Cmuig'** |
| s[0::4] | **'Cug'** |
| s[9:0:-2] | **'gium'** |

6.

|  |  |
| --- | --- |
| **Code** | **Extracted String** |
| s[0:6] | Imagin |
| s[5:11] **OR** s[5:] | nation |
| s[::3] | Igao |

7.

|  |  |
| --- | --- |
| √ | z[:6] |
| √ | z[::3] |
| √ | z[-1:10:-1] |
| √ | z[:10:-1] |
|  | z[0:10:-1] |
|  | z[:-6:-1] |
|  | z[-1:-7] |
| √ | z[-6:][::-1] |
| √ | z[0:2] + (z[-2] + z[2])[::-1] \* 2 |
| √ | z[:-11] |

8.

|  |  |  |
| --- | --- | --- |
| **Escape Code** | **Code** | **What the escape code does** |
| \n | >>> **print("Phrase 1.\n Phrase 2.")** | Prints on a new line. |
| \t | >>> **print("Phrase 1.\t Phrase 2.")** | Prints a tab space. |

9. Actual Output:

|  |
| --- |
| **Let’s try escape characters.**  **Type them out.**  **How will it appear?** |

1. Corrected Code:

|  |
| --- |
| **'2017 is Bala\'s first year of studying Computing.'** |

1. Actual Output:

|  |
| --- |
| **There are 2 types of people in this world.**  **Males from Mars and females from Venus** |

12. Corrected Code:

|  |
| --- |
| **user\_name = input("Enter your name please: ")** |

1. Corrected Code:

|  |
| --- |
| **value = int(input("Enter your age: "))** |

1. Refer to 10q14.py

|  |
| --- |
| **What is your name? John**  **Hello, John**  **How are you today?** |

**Note:** The output will vary according to your inputs.

1. Refer to 10q15.py

|  |
| --- |
| **What is your favourite movie? The Avengers**  **How old are you? 15**  **At age 15, your favourite movie is The Avengers.** |

**Note:** The output will vary according to your inputs.

1. Refer to 10q16.py

|  |
| --- |
| **# This program calculates body mass index**  **weight = float(input("Enter your weight in kg: "))**  **height = float(input("Enter your height in metres: "))**  **bmi = weight/(height \*\* 2)**  **print("Your body mass index is {}.".format(bmi))** |

|  |  |  |  |
| --- | --- | --- | --- |
| **Worksheet 11** | **4.1** | **Program Development** | **User-defined functions** |

1. Actual Output of *print\_hello.py*

|  |
| --- |
| Hello, World!  None |

2. Actual Output of *return\_hello.py*

|  |
| --- |
| Hello, World! |

3. For print\_hello.py,

* Running hello() will output 'Hello World!'
* Since there is no return value, the original function call to hello() will be replaced with 'None'.
* Printing the return value will thus output 'None'

For return\_hello.py.

* Running hello() will return 'Hello World!'
* The original function call to hello() will be replaced with 'Hello World!'.
* Printing the return value will thus output 'Hello World'

4. Actual Output of *print\_outside.py*

|  |
| --- |
| 19  65 |

5. Actual Output of *print\_within.py*

|  |
| --- |
| 20  67 |

6. For print\_outside.py, global variables num1 and num2 are printed.

For print\_within.py, local variables num1 and num2 are printed.

7. Actual Output of *sample1.py*

|  |
| --- |
| 1235 |

8. Actual Output of *sample2.py*

|  |
| --- |
| UnboundLocalError: local variable 'num1' referenced before assignment |

9. Functions have only read access but not write access to global variables thus num1 will be treated as local variable.

10. A possible solution is:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7 | def multiply(num1, num2):  answer = num1 \* num2  return answer    number1 = 19  number2 = 65  print(multiply(number1, number2)) |

|  |  |  |  |
| --- | --- | --- | --- |
| **Worksheet 12** | **4.1** | **Program Development** | **Selection Statements** |

|  |  |  |  |
| --- | --- | --- | --- |
| **Part** | **Code** | **Number of Branches that Produce Output** | **Actual output** |
| a | **a = 7**  **if a > 0:**  **print('a is positive!')** | 1 | **a is positive!** |
| b | **a = -7**  **if a > 0:**  **print('a is positive!')**  **else:**  **print('a is negative or zero!')** | 2 | **a is negative or zero!** |
| c | **a = 0**  **if a > 0:**  **print('a is positive!')**  **elif a == 0:**  **print('a is zero!')**  **else:**  **print('a is negative!')** | 3 | **a is zero!** |

1. Suggested Code:

|  |
| --- |
| **x = int(input("How old will you be as of 1 Mar? "))**  **if x < 21:**  **print("Sorry, you can't vote yet!")**  **else:**  **print("Hurray, you are eligible for voting!")**  **print("The age to vote is 21 years or older. ")** |

1. The if statement has 5 branches. The program takes in an input from the user, which is a chemical formula, and outputs the corresponding chemical name.
2. The 2 syntax errors are: (a) there is a missing right parenthesis in line 1 and (b) the relational operator in line 2 should be '>=' without any space in between.

Suggested Code:

|  |
| --- |
| **temperature = float(input("Enter today's temperature: "))**  **if temperature >= 30:**  **print("The weather is warm today.")**  **else:**  **print("The weather is less than 30 degrees Celsius today.")** |

1. The '==' in line 1 of Block A is incorrect and should be circled.

The first line of code should be an assignment statement using = where variable a is assigned some value. However, in Block A the first line is performing an equality check using == instead, which is incorrect because the variable a has no assigned value yet at that part of the program.

1. The error is in line 2 '**if temperature > 29:**' because the first range overlaps with the second range. For input values 30 to 32, the first message will always be printed instead of the second message because the first condition is always tested to select the first branch and the second condition is skipped. Hence the expected output of **"It is moderately hot today"** will not be printed but **"What a hot day!"** will be printed instead. This is an example of a logic error where the program does not give the expected output.

Corrected Code:

|  |
| --- |
| **temperature = int(input("What is the temperature today? "))**  **if temperature > 32:**  **print("What a hot day!")**  **elif temperature >= 28 and temperature <= 32:**  **print("It is moderately hot today.")**  **else:**  **print("It is cooling today.")** |

1. D
2. Refer to 12q8.py.

|  |
| --- |
| **g = 9.8**  **m = float(input("Enter the object's mass in kg: "))**  **h = float(input("Enter the object's height in m: "))**  **pe = m \* g \* h**  **print("The object has " + str(pe) +**  **" J of gravitational potential energy.")** |

1. Refer to 12q9.py

|  |
| --- |
| **x = int(input("Enter a PSI value :"))**  **if x <= 50:**  **print("The PSI level is good.")**  **elif x > 50 and x <= 100:**  **print("The PSI Level is moderate.")**  **elif x > 100 and x <= 200:**  **print("The PSI Level is unhealthy.")**  **elif x > 200 and x <= 300:**  **print("The PSI Level is very unhealthy.")**  **else:**  **print("The PSI Level is hazardous.")** |

1. Refer to 12q10.py

|  |
| --- |
| **sides = int(input("Enter number of regular polygon sides (3 to 8): "))**  **while sides < 3 or sides > 8:**  **sides = int(input("Enter number of regular polygon sides (3 to 8): "))**  **if sides == 3:**  **print("This is a triangle.")**  **elif sides == 4:**  **print("This is a square.")**  **elif sides == 5:**  **print("This is a pentagon.")**  **elif sides == 6:**  **print("This is a hexagon.")**  **elif sides == 7:**  **print("This is a heptagon.")**  **else:**  **print("This is an octagon.")** |

1. There are 2 possible solutions – using selection or iteration. (Prompt: which solution is better?)

Refer to 12q11\_selection.py.

|  |
| --- |
| **# Solution 1: Use of selection**  **count\_neg = 0**  **count\_zero = 0**  **count\_pos = 0**  **x1 = int(input("Enter an integer: "))**  **x2 = int(input("Enter an integer: "))**  **x3 = int(input("Enter an integer: "))**  **x4 = int(input("Enter an integer: "))**  **x5 = int(input("Enter an integer: "))**  **x6 = int(input("Enter an integer: "))**  **x7 = int(input("Enter an integer: "))**  **x8 = int(input("Enter an integer: "))**  **x9 = int(input("Enter an integer: "))**  **x10 = int(input("Enter an integer: "))**  **if x1 < 0:**  **count\_neg = count\_neg + 1**  **elif x1 == 0:**  **count\_zero = count\_zero + 1**  **else:**  **count\_pos = count\_pos + 1**  **if x2 < 0:**  **count\_neg = count\_neg + 1**  **elif x2 == 0:**  **count\_zero = count\_zero + 1**  **else:**  **count\_pos = count\_pos + 1**  **if x3 < 0:**  **count\_neg = count\_neg + 1**  **elif x3 == 0:**  **count\_zero = count\_zero + 1**  **else:**  **count\_pos = count\_pos + 1**  **if x4 < 0:**  **count\_neg = count\_neg + 1**  **elif x4 == 0:**  **count\_zero = count\_zero + 1**  **else:**  **count\_pos = count\_pos + 1**  **if x5 < 0:**  **count\_neg = count\_neg + 1**  **elif x5 == 0:**  **count\_zero = count\_zero + 1**  **else:**  **count\_pos = count\_pos + 1**  **if x6 < 0:**  **count\_neg = count\_neg + 1**  **elif x6 == 0:**  **count\_zero = count\_zero + 1**  **else:**  **count\_pos = count\_pos + 1**  **if x7 < 0:**  **count\_neg = count\_neg + 1**  **elif x7 == 0:**  **count\_zero = count\_zero + 1**  **else:**  **count\_pos = count\_pos + 1**  **if x8 < 0:**  **count\_neg = count\_neg + 1**  **elif x8 == 0:**  **count\_zero = count\_zero + 1**  **else:**  **count\_pos = count\_pos + 1**  **if x9 < 0:**  **count\_neg = count\_neg + 1**  **elif x9 == 0:**  **count\_zero = count\_zero + 1**  **else:**  **count\_pos = count\_pos + 1**  **if x10 < 0:**  **count\_neg = count\_neg + 1**  **elif x10 == 0:**  **count\_zero = count\_zero + 1**  **else:**  **count\_pos = count\_pos + 1**  **print("The total number of negative integers are: " + str(count\_neg))**  **print("The total number of zeroes are: " + str(count\_zero))**  **print("The total number of positive integers are: " + str(count\_pos))** |

Refer to 12q11\_iteration.py.

|  |
| --- |
| **# Solution 2: Use of iteration loops**  **count\_neg = 0**  **count\_zero = 0**  **count\_pos = 0**  **for i in range(10):**  **x = int(input("Enter an integer :"))**  **if x < 0:**  **count\_neg = count\_neg + 1**  **elif x == 0:**  **count\_zero = count\_zero + 1**  **else:**  **count\_pos = count\_pos + 1**  **print("The total number of negative integers are: " + str(count\_neg))**  **print("The total number of zeroes are: " + str(count\_zero))**  **print("The total number of positive integers are: " + str(count\_pos))** |

1. Refer to 12q12.py

|  |
| --- |
| **menu = "**\*\*\*\*\*Types of operation\*\*\*\*\***\n\**  **1. +\n\**  **2. -\n\**  **3. \*\n\**  **4. /\n\**  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n**"**  **print(menu)**  **op = int(input("Key in the number for operator (1 to 4): "))**  **num1 = int(input("Enter your first number? "))**  **num2 = int(input("Enter your second number? "))**  **if op == 1:**  **print(num1 + num2)**  **elif op == 2:**  **print(num1 - num2)**  **elif op == 3:**  **print(num1 \* num2)**  **elif op == 4:**  **print(num1 / num2)** |

|  |  |  |  |
| --- | --- | --- | --- |
| **Worksheet 13** | **4.1** | **Program Development** | **Loops** |

|  |  |  |
| --- | --- | --- |
|  | **Code** | **Actual Output** |
| (a) | **for x in range(10):**  **print(x)** | 0  1  2  3  4  5  6  7  8  9 |
| (b) | **for x in range(6,10):**  **print(x)** | 6  7  8  9 |
| (c) | **for x in range(1,5):**  **print(x,"pizza slice")** | 1 pizza slice  2 pizza slice  3 pizza slice  4 pizza slice |
| (d) | **for x in range(2,9,2):**  **print(x)** | 2  4  6  8 |
| (e) | **fruits = ['apples', 'oranges',**  **'bananas', 'grapes']**  **for fruit in fruits:**  **print("Fruit: {}".format(fruit))** | Fruit: apples  Fruit: oranges  Fruit: bananas  Fruit: grapes |

|  |  |  |
| --- | --- | --- |
|  | **Code** | **Actual Output** |
| (a) | **a = 0**  **while a < 10:**  **a = a + 1**  **print(a)** | 1  2  3  4  5  6  7  8  9  10 |
| (b) | **x = 10**  **while x != 0:**  **print(x)**  **x = x - 1**  **print("Counting down ...")** | 10  9  8  7  6  5  4  3  2  1  Counting down ... |
| (c) | **x = input("Enter a countdown number \**  **between 1 to 20. ")**  **x = int(x)**  **while x != 0:**  **print(x)**  **x = x - 1**  **print("Countdown completed!")** | 20  19  18  17  16  15  14  13  12  11  10  9  8  7  6  5  4  3  2  1  Countdown completed!  **Note:** The countdown will start from the number you input. |

1. What happened? The loop doesn’t end and the output is always for “Register number 1”.

State a possible reason why this happened. The counter was not increased after each iteration.

Amendment: Line number 12  
 Code: **counter += 1**

1. A possible answer is as follows:

Error 1 is on line 3

Correction: **for i in range(4):**

Error 2 is on line 4

Correction: **x = int(input("Enter a number: "))**

Error 3 is on line 5

Correction: **total = total + x**

Error 4 is on line 6

Correction: **print("The total is:", total)**

1. Actual Output:

|  |
| --- |
| 1 x 1 = 1  1 x 2 = 2  1 x 3 = 3  2 x 1 = 2  2 x 2 = 4  2 x 3 = 6  3 x 1 = 3  3 x 2 = 6  3 x 3 = 9 |

1. A possible answer is as follows:

|  |
| --- |
| **x = 20**  **while x >= 0:**  **print(x)**  **x = x - 1**  **print("End of while loop")** |

1. A possible answer is as follows:

|  |
| --- |
| **x = int(input("Enter a positive integer: "))**  **while x > 0:**  **print("Siti")**  **x = x - 1**  **print("Completed")** |

**Note:** “Siti” should be replaced with student’s name.

1. A possible answer is as follows:

|  |
| --- |
| **x = int(input("What multiplication table do you want? "))**  **for y in range(1, 11):**  **print(x, "x", y, "=", x \* y)** |

1. A possible answer is as follows:

|  |
| --- |
| **x = int(input("Enter a positive integer: "))**  **factorial = 1**  **for x in range(1, x + 1):**  **factorial = factorial \* x**  **print("Factorial: ", factorial)** |

1. A possible answer is as follows:

|  |  |
| --- | --- |
| 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 | **# Input isbn = input("Enter first 9 digits of ISBN: ")  # Process check\_sum = 0 for index in range(9):  check\_sum += (10 - index) \* int(isbn[index]) remainder = check\_sum % 11 if remainder == 0:  check\_digit = 0 else:  check\_digit = 11 - remainder  if check\_digit == 10:  check\_digit = "X"**  **# Output print(check\_digit)** |

|  |  |  |  |
| --- | --- | --- | --- |
| **Worksheet 14** | **4.1** | **Program Development** | **Data Validation, Test Cases, Program Errors, Debugging** |

1. Suggested Code:

|  |
| --- |
| **while True:**  **input\_str = input("Enter an integer: ")**  **digits = input\_str**  **if len(digits) > 0 and digits[0] == '-':**  **digits = digits[1:]**  **if len(digits) > 0 and digits.isdigit():**  **break**  **print("Data validation failed!")**  **number = int(input\_str)**  **print(number \* 2)** |

1. Suggested Code:

|  |
| --- |
| **is\_valid = False**  **while not is\_valid:**  **input\_str = input("Enter a phone number: ")**  **is\_valid = len(input\_str) == 9 and input\_str[:4].isdigit() \**  **and input\_str[4] == '-' and input\_str[5:].isdigit()** |

|  |  |
| --- | --- |
| **Test Case 1 for Normal Conditions** | |
| **Expected Input** | 2017 |
| **Expected Output** | 10 |

|  |  |
| --- | --- |
| **Test Case 2 for Normal Conditions** | |
| **Expected Input** | 1965 |
| **Expected Output** | 21 |

|  |  |
| --- | --- |
| **Test Case 1 for Error Conditions** | |
| **Expected Input** | -2017 |
| **Expected Output** | Data validation failed!  Input must be positive |

|  |  |
| --- | --- |
| **Test Case 2 for Error Conditions** | |
| **Expected Input** | seventeen |
| **Expected Output** | Data validation failed!  Input must be an integer |

|  |  |
| --- | --- |
| **Test Case 1 for Boundary Conditions** | |
| **Expected Input** | 00:00 |
| **Expected Output** | There are 1440 minutes left until the next midnight. |

|  |  |
| --- | --- |
| **Test Case 2 for Boundary Conditions** | |
| **Expected Input** | 23:59 |
| **Expected Output** | There is 1 minute left until the next midnight. |

|  |  |
| --- | --- |
| **Test Case 3 for Boundary Conditions** | |
| **Expected Input** | 24:00 |
| **Expected Output** | Data validation failed!  Invalid time |

|  |  |
| --- | --- |
| **Type of Error** | **Description** |
| Syntax errors | * Detected by translators, such as:   + **Compilers**, which store the translated code (or machine code) on the computer such that the translator (or compiler) and source code are no longer needed to run the program, or   + **Interpreters**, which translate the source code while the program is running so both the translator (or interpreter) and source code are still needed to run the program * Caused by spelling mistakes or the incorrect sequence (or combination) of symbols in source code |
| Run-time errors | * Occur while the program is running and may cause the program to crash or hang * Caused by:   + Incorrect use of commands (or functions)   + Input data that has not been properly validated   + Conditions outside of the program’s control |
| Logic errors | * Detected when the actual output of a program does not match the expected output * Caused by use of an incorrect or incomplete algorithm |

1. What character is responsible for the discrepancy? ' ' *(space)*

Suggested Code:

|  |
| --- |
| **message = input("Enter a message: ")**  **punctuation\_count = 0**  **for character in message:**  **if not (character.isalnum() or character.isspace()):**  **punctuation\_count += 1**  **print(punctuation\_count)** |

1. Explanation why program is incorrect:

**x = y** assigns the value of 5 to the variable x while the previous value of 3 is overwritten. However, immediately following this assignment, **y = x** assigns the value of x to y, which is 5 in this case. Hence, both variables x and y now have the same value of 5.

Suggested Code:

Use a third variable to swap the values of 2 variables.

|  |
| --- |
| **x = 3**  **y = 5**  **temp = x**  **x = y**  **y = temp** |

1. What is the value of part1 when the output starts to differ (for input 10)? 8

What is the value of part2 when the output starts to differ (for input 10)? 1.0

|  |  |  |  |
| --- | --- | --- | --- |
| **Worksheet 15** | **1.3** | **Ethical, Social and Economic Issues** | **Data Safety**  **Social and Economic Issues**  **Ethical Issues** |

1. Match the preventive measure to the type of cyberattack.

|  |  |  |  |
| --- | --- | --- | --- |
| **Preventive measure** |  |  | **Type of cyberattack** |
| Install software to scan computer and update virus definitions regularly |  |  | Pharming |
|  |  |  |  |
| Opt out of subscribing to mailing lists that are not needed |  |  | Phishing |
|  |  |  |  |
| Do not click on hyperlinks in emails from untrusted sources |  |  | Spam |
|  |  |  |  |
| When visiting a website, look out for a padlock icon. |  |  | Virus |

1. Phishing refers to the use of emails to steal personal information from users. These details could then be used to commit a range of frauds such as buying goods or services using the details or even taking out loans by forging a person’s identity.
2. Some ways to avoid being phished include:

* Communicate personal information in person or through secured web sites only.
* Do not click on links, download files or open attachments in emails from unknown sources.
* Check online accounts and bank statements regularly to ensure that no unauthorized transactions have been made.

1. Encryption refers to the encoding of data so that even if it were intercepted, the person would be unable to read or understand the encrypted data without the use of a secret key.
2. Encryption is needed to ensure the privacy and security of bank account and other personal details.
3. Any 3 of the following:

* Passwords should be kept safe and secure
* Avoid obvious password choices
* Use passwords that are a mixture of lower-case letters, upper-case letters, numbers and symbols
* Avoid the reuse of passwords
* Update passwords regularly

1. Data protection rules that are True:

|  |  |
| --- | --- |
| √ | If you are approached on the street to complete a survey form, you are deemed to have consented to the use of your personal data for that specific purpose. |
|  | If you have provided your personal data to an organisation, you are not allowed to withdraw your consent to the use by giving notice to that organisation. |
| √ | Organisations that collect personal data are required to obtain the consent of the user with regard to the collection, use and disclosure of the data. |
|  | You cannot write in to an organisation to request them to correct an error in your personal data. |

1. Working from home by using technology to support communication and work.
2. VoIP that uses an existing Internet connection for calls is cheaper than installing a physical phone line.
3. Advantages and disadvantages
4. Any 2 of the following:
   * Less time is spent on commuting for meetings
   * Meetings can be organised and held at any time
   * People do not have to spend time away from their families
5. Any 2 of the following:

* Face to face contact is still considered important for meetings
* People may not be able to focus so well on work if they work from home
* May be difficult to arrange a meeting when people are in places of different time zones

1. Hacking is gaining unauthorized access to a computer system or network. Hackers can gain access to important data used for terrorism and organised crime that could lead to national security issues or data loss for individuals or organisations.
2. Firewalls monitor each packet of data that arrives at a network and will either block or allow it to pass through based on a set of configured rules. They work to prevent unauthorised access by rejecting any data that fail to pass the authorization criteria specified by the rules.

|  |  |  |  |
| --- | --- | --- | --- |
| **Worksheet 16** | **1.2** | **Data Representation** | **Number Systems** |

1.

1. (0110 1100)2
2. (6C)16
3. Accept any valid answer (e.g., MAC addresses and RGB codes)

2.

1. There are faults in the display and motor subsystems for elevator number 7.
2. There are faults in the motor and ventilator subsystems for elevator number 11.
3. 11001110

3.

1. Electrical point number 56 is operational.
2. 00100100
3. (9C)16

|  |  |  |  |
| --- | --- | --- | --- |
| **Worksheet 17** | **2.1** | **Computer Architecture** | **Boolean Logic and Logic Gates** |

1.

|  |  |
| --- | --- |
| **Logic States** | |
| 0 | **1** |
| **False** | True |
| **Off** | On |
| Low | **High** |

1. a) OR b) AND c) NOT

a)

|  |  |
| --- | --- |
| **Input** | **Output** |
| *Rain? (True/False)* | *Umbrella needed? (True/False)* |
| False | False |
| True | True |

b)

|  |  |  |
| --- | --- | --- |
| **Inputs** | | **Output** |
| *Is late for school? (True/False)* | *Has at least $10? (True/False)* | *Will take taxi to school? (True/False)* |
| False | False | False |
| False | True | False |
| True | False | False |
| True | True | True |

|  |  |
| --- | --- |
| **Logic Gate** | **Boolean Statement** |
| AND | Q = X AND Y |
| OR | Q = X OR Y |
| NAND | Q = NOT (X AND Y) |
| NOR | Q = NOT (X OR Y) |

1. AND gate b) OR gate

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Inputs** | | **Output** |  | **Inputs** | | **Output** |
| **X** | **Y** | **Q** |  | **X** | **Y** | **Q** |
| 0 | 0 | 0 |  | 0 | 0 | 0 |
| 0 | 1 | 0 |  | 0 | 1 | 1 |
| 1 | 0 | 0 |  | 1 | 0 | 1 |
| 1 | 1 | 1 |  | 1 | 1 | 1 |

1. NAND gate d) NOR gate

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Inputs** | | **Output** |  | **Inputs** | | **Output** |
| **X** | **Y** | **Q** |  | **X** | **Y** | **Q** |
| 0 | 0 | 1 |  | 0 | 0 | 1 |
| 0 | 1 | 1 |  | 0 | 1 | 0 |
| 1 | 0 | 1 |  | 1 | 0 | 0 |
| 1 | 1 | 0 |  | 1 | 1 | 0 |

6.

1. Diagram:

Switch

Alarm

Motion sensor

1. Truth Table:

|  |  |  |
| --- | --- | --- |
| **Inputs** | | **Output** |
| **Switch** | **Motion Sensor** | **Alarm** |
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

The alarm will only sound when the switch is on and the motion sensor detects movement.

|  |  |  |  |
| --- | --- | --- | --- |
| **Worksheet 18** | **2.1** | **Computer Architecture** | **Logic Circuits and Truth Tables** |

1.

1. Q = A OR (NOT B)

|  |  |  |  |
| --- | --- | --- | --- |
| **A** | **B** | **NOT B** | **Q** |
| 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 0 |
| 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 1 |

1. Q = (A AND B) OR C

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **A** | **B** | **C** | **A AND B** | **Q** |
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 1 |
| 0 | 1 | 0 | 0 | 0 |
| 0 | 1 | 1 | 0 | 1 |
| 1 | 0 | 0 | 0 | 0 |
| 1 | 0 | 1 | 0 | 1 |
| 1 | 1 | 0 | 1 | 1 |
| 1 | 1 | 1 | 1 | 1 |

1. Q = NOT (A OR B) AND C

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **A** | **B** | **C** | **A OR B** | **NOT (A OR B)** | **Q** |
| 0 | 0 | 0 | 0 | 1 | 0 |
| 0 | 0 | 1 | 0 | 1 | 1 |
| 0 | 1 | 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 1 | 0 | 0 |
| 1 | 0 | 0 | 1 | 0 | 0 |
| 1 | 0 | 1 | 1 | 0 | 0 |
| 1 | 1 | 0 | 1 | 0 | 0 |
| 1 | 1 | 1 | 1 | 0 | 0 |

2.

|  |  |  |  |
| --- | --- | --- | --- |
| **Ignition** | **Seat Belt** | **Engine**  **(Ignition AND Seat Belt)** | **Buzzer**  **(Ignition AND NOT Seat Belt)** |
| 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 0 |
| 1 | 0 | 0 | 1 |
| 1 | 1 | 1 | 0 |

3. a) There are two inputs: A (temperature sensor) and B (motion sensor)

|  |  |  |
| --- | --- | --- |
| **Inputs** | | **Output** |
| **A** | **B** | **Q** |
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

This gives the truth table of an AND gate.

b) Logic Circuit Diagram

A: Temperature Sensor

Q: Aircon Status

B: Motion Sensor

4. a)

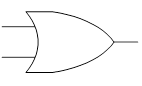


A: Understood lessons in class

B: Consulted teacher on lessons

C: on MC

Z: Could complete homework



b)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **A** | **B** | **C** | **A OR B** | **NOT C** | **Z** |
| 0 | 0 | 0 | 0 | 1 | 0 |
| 0 | 0 | 1 | 0 | 0 | 0 |
| 0 | 1 | 0 | 1 | 1 | 1 |
| 0 | 1 | 1 | 1 | 0 | 0 |
| 1 | 0 | 0 | 1 | 1 | 1 |
| 1 | 0 | 1 | 1 | 0 | 0 |
| 1 | 1 | 0 | 1 | 1 | 1 |
| 1 | 1 | 1 | 1 | 0 | 0 |

5. a) Diagram:

A: Front door

B: Side door

Z: Alarm

D: Keypad



C:

b) Boolean statement:

Z = (A OR B) AND D

c) Truth Table:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **A** | **B** | **D** | **C** | **Z** |
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 0 |
| 0 | 1 | 0 | 1 | 0 |
| 0 | 1 | 1 | 1 | 1 |
| 1 | 0 | 0 | 1 | 0 |
| 1 | 0 | 1 | 1 | 1 |
| 1 | 1 | 0 | 1 | 0 |
| 1 | 1 | 1 | 1 | 1 |

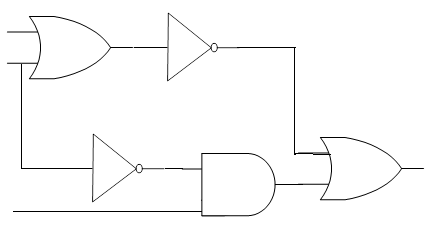
**Note:** The keypad, D, is switched on when the owner is not at home. Hence, there are only 3 situations where the alarm, Z, will be triggered:

* when the owner is not at home (D=1) and only front door is opened (A=1)
* when the owner is not at home (D=1) and only side door is opened (B=1)
* when the owner is not at home (D= 1) and both front (A=1) and side doors (B=1) are opened

6. a) The truth table is:

|  |  |  |  |
| --- | --- | --- | --- |
| **Inputs** | | | **Output** |
| **T** | **C** | **M** | **L** |
| 0 | 0 | 0 | 1 |
| 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 0 |
| 1 | 1 | 1 | 0 |

b) The logic circuit is:



T

C

M

L

[accept answer if student did not simplify]

c) The Boolean statement is:

L = (NOT T AND NOT C) OR (M AND NOT C)

= NOT (T OR C) OR (M AND NOT C)

7 a) Boolean Statement

Q = (A AND NOT B AND C) OR (NOT A AND B AND NOT C)

b) Logic circuit diagram

A

B

C

Q

c) Truth table

|  |  |  |  |
| --- | --- | --- | --- |
| **Inputs** | | | **Output** |
| **A** | **B** | **C** | **Q** |
| 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 1 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 0 |
| 1 | 1 | 1 | 0 |

|  |  |  |  |
| --- | --- | --- | --- |
| **Worksheet 19** | **1.1** | **Data Management** | **Data Tabulation, Conditional Statements** |

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Column A** | **Column B** | **Column C** |
| **Column Heading** | Date | Subject Name | Amount Spent |
| **Data Type** | Date | Text | Currency |

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Column A** | **Column B** | **Column C** |
| **Column Heading** | Greeting Phrase | Price | Discount |
| **Data Type** | Text | Currency | Percentage |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Column A (Item)** | **Column B (Quantity)** | **Column C (Unit)** | **Column D (Unit Price)** | **Column E (Subtotal)** |
| **Data Type** | Text | Number | Text | Currency | Currency |

1. FALSE
2. TRUE
3. TRUE
4. = IF(AND(C2>=40, C2<=50), "Yes", "No")
5. = AND(E2="Yes", F2="Yes")
6. = IF(AND(F2="Yes", NOT(D2)), "Yes", "No")

|  |  |  |  |
| --- | --- | --- | --- |
| **Worksheet 20** | **1.1** | **Data Management** | **Mathematical and Statistical Functions** |

1. a) 4

b) 5

c) 1

d) 0

e) -3

2. a) 3

b) 4

c) 0

d) -1

e) -4

3. a) 30

b) 5

c) -0.23

d) 1000

e) 1

f) 0

g) -100

4. a) =(-B2+SQRT(B2^2-4\*B1\*B3))/(2\*B1)

b) =(-B2-SQRT(B2^2-4\*B1\*B3))/(2\*B1)

5. a) =RAND()\*(70-40)

b) =C2+40 **OR** =RAND()\*(70-40)+40

6. a) =RANDBETWEEN(1, 6)

b) =C8+H8+M8

7. Refer to solution in source files.

8. a) =SUMIF(C2:C15, "F", D2:D15)

b) =COUNTIF(C2:C15,"F")

c) =G2/G3

d) =SUMIF(C2:C15, "M",D2:D15)

e) = COUNTIF(C2:C15, "M")

f) =G5/G6

g) =MIN(D2:D15)

h) =SMALL(D2:D15,2)

i) =MAX(D2:D15)

j) =LARGE(D2:D15,2)

k) =MODE.SNGL(D2:D15)

l) =MEDIAN(D2:D15)

m) =AVERAGE(D2:D15)

9. a) =COUNTA(D2:H2)

b) =SUM(I2:I10)

c) =COUNTBLANK(D2:D10)

|  |  |  |  |
| --- | --- | --- | --- |
| **Worksheet 21** | **1.1** | **Data Management** | **Lookup, Date and Text Functions** |

1. =HLOOKUP(B6, $B$25:$E$26, 2, FALSE)

2. =VLOOKUP(B6, $A$26:$B$29, 2, FALSE)

3. Refer to solution in source files.

4. Refer to solution in source files.

5. Refer to solution in source files.

6. Refer to solution in source files.

7. a) =MID(A1, 5,4)

b) =LEFT(A1,1)

8. a) False

b) True

c) True

|  |  |  |  |
| --- | --- | --- | --- |
| **Worksheet 21** | **2.2** | **Data Communications** | **Networks** |

1. Any two from:

* Shared resources: A network allows a group of computers to make use of shared resources such as printers or files.
* Shared Internet access: Depending on the network’s configuration, every user who logs on to the network may have access to the Internet.
* Shared software: Software can be stored on the central server of a network and deployed to other computers over a network.
* Shared storage: Data files can be stored on a central server for ease of access and backup purposes.
* Communication: Computers in the same network are often able to share instant messages and emails for communication.

1. Any two from:

* Security risks: As files are shared through a network, there is the risk of virus or worm attacks spreading throughout the network even with just one infected computer.
* Risk of data loss: Data may become lost due to hardware failures or errors. Using a network means regular data backups are needed.
* Server outage: If the server fails, the network will not be able to function, thus affecting work processes.

1. LAN

* Network of computing devices connected within a small geographical area, typically within the same building, such as a home, school or office
* Provide faster data transfer than other networks

MAN

* Network of computing devices typically spanning across two or more buildings within the same town or city

WAN

* Network of computing devices covering a large-scale geographical area, typically across multiple geographical locations

1. (a) MAN
2. LAN
3. WAN
4. (a) Radio waves
5. Advantage (Any one from)

* For wireless network, mobility of users is higher as users can move about freely within the range of the wireless network. For wired networks, mobility of users is lower as network connections are fixed at specific spots and users cannot move to other locations.
* For wireless network, adding new devices to the network is easier as the router can be easily configured. For wired networks, it is more cumbersome to add new devices to the network as physical constraints and the running of cables need to be considered.
* Wireless networks are more organised without cables. Wired networks tend to look more disorganised due to cables running across floors.

Disadvantage (Any one from)

* For wireless networks, transfer rate is generally slower and has lower bandwidth due to possible interference from radio waves or microwaves; varies according to user location in relation to network. For wired network, transfer rate is faster and has higher bandwidth as cables provide dedicated connection.
* Wireless networks are less reliable due to potential interference from radio waves and microwaves or blockage from physical obstructions. Wired networks are more reliable as data transmission is unaffected by interference.
* Wireless networks are less secure due to possible intrusion by hackers. Wired networks are more secure as the network is less susceptible to interception and hacking.

1. Advantage (any one from)

* For client-server networks, centralised control of data and resources makes it easier to schedule backups of all shared files at regular intervals as compared to P2P networks where data and resources are stored locally within each computer.
* For client-server networks, security may be enhanced as access rights can be controlled centrally at a server as compared to P2P networks where security is handled by each computer.

Disadvantage (any one from)

* For client-server networks, there is a higher setup cost due to the need for a higher-performance server. For P2P networks, the setup cost is lower as basic computers can act as servers.
* For client-server networks, there is administrative costs needed for the maintenance of server and clients which will not be incurred for P2P networks.

1. Ethernet
2. Spot Printers uses a client-server network model where one or more servers provides access to shared data, hardware (i.e., printers) and resources at the request of clients. There is also a clear distinction of which computers are clients and which computers are servers.
3. B
4. C. The MAC address of a NIC is stored in a unique 48-bit string. Each 8-bit/byte is represented as two hexadecimal digits.
5. You would be responsible for all devices found internally within the company network:

|  |  |
| --- | --- |
| √ | Desktop computers |
| √ | Router |
|  | Email server |
| √ | File and print servers |

12.

|  |  |  |
| --- | --- | --- |
| **ASCII character** | **Data received at destination** | **Is there an error?** |
| t | 01110100 | No |
| e | 11100101 | Yes |
| a | 11100001 | No |