|  |
| --- |
|  |

|  |
| --- |
| **QA Consulting.** |
| What is Programming? |
| exercise guide 1 |

Exercise 1.1 – Program Development Lifecycle

## Please watch either or both of the videos:

* <https://www.youtube.com/watch?v=YV6ykfG1nQY>
* <https://www.youtube.com/watch?v=Sud_euZdNlk>

## Make notes and create a short presentation that:

* List the steps of the Program Development Lifecycle
* Briefly describe the steps and key activities
* Highlights how do the two life cycles differ?
* Includes any other stages that could be added?

# Exercise 1.2 – Simple Sequence

Please design and write a program on “paper” that:

* Inputs the value of two cards
* Multiplies them together
* Outputs the result.

Note:

* You will need a means of storing the value of both cards and the result.

Be prepared to share your solution with the class.

# Exercise 1.3 – Simple If

Please design and write a program on “paper” that:

* Inputs the value of two cards
* If the value of the second card is greater than 5 add its value to the value of the first card
* Outputs the “value” of the first card

Note:

* You will need a means of storing the value of both cards

Be prepared to share your solution with the class.

# Exercise 1.4 – If…Else

Please design and write a program on “paper” that:

* Inputs the value of two cards and an operator
* The operator must be a + or \*
* Using that operator perform the appropriate calculation on the two values
* Outputs the result

Note:

* You will need a means of storing the value of both cards, the operator and the result

Be prepared to share your solution with the class.

# Exercise 1.5 – Case

Please design and write a program on “paper” that:

* Inputs the value of two cards and an operator
* The operator can be a +, -, \* or /
* Using that operator perform the appropriate calculation on the two values
* Outputs the result

Note:

* You will need a means of storing the value of both cards, the operator and the result
* Can use repeated ifs / else-ifs or selection (case) statement

Be prepared to share your solution with the class.

# Exercise 1.6 – Simple Loop

Please design and write a program on “paper” that:

* Inputs the value of the first card
* Processes that number of cards inputting each value
* Calculates the sum of the values
* Outputs the result

Note

* You will need a means of storing the value of the first card, next card, count of cards processed and the result

Be prepared to share your solution with the class.

# Exercise 1.7 – Loop with Exit

Please design and write a program on “paper” that

* Processes the pack of cards inputting each value until a heart is reached
* Calculates the sum the values
* Outputs the result

Note

* You will need a means of storing the value and suit of each cards and the result

Be prepared to share your solution with the class.

# Exercise 1.8 – Loop with Exit and Case

Please design and write a program on “paper” that

* Processes the cards in the pack
* Counts the number of cards for each suit
* Continues until one of the counts reaches five
* Outputs each suit and the count for that suit

Note:

* You will need a means of storing the suit of the next card and the count for each suit

Be prepared to share your solution with the class.

# Solutions

## Solution 1.1 – Program Development Lifecycle

See Delegate Guide slides and notes.

## Solution 1.2 – Simple Sequence

Pseudo Code:

* Output “Please enter value of first card:”
* Enter value of first card and store in Card1
* Output “Please enter value of second card:”
* Enter value of second card and store in Card1
* Calculate Card1 x Card2 and store in Result
* Output “Result” and value of Result

## Solution 1.3 – Simple If

Pseudo Code:

* Output “Please enter value of first card:”
* Enter value of first card and store in Card1
* Output “Please enter value of second card:”
* Enter value of second card and store in Card1
* If Card2 > 5
  + Calculate Card1 + Card2 and store in Card1
* Output “Card1” and value of Card1

## Solution 1.4 – If…Else

Pseudo Code:

* Output “Please enter value of first card:”
* Enter value of first card and store in Card1
* Output “Please enter value of second card:”
* Enter value of second card and store in Card1
* Output “Please enter operator + or \*:”
* Enter value of operator and store in Operator
* If Operator = +
  + Calculate Card1 + Card2 and store in Result
* Else
  + Calculate Card1 \* Card2 and store in Result
* Output “Result:” and value of Result

## Solution 1.5 – Case

Pseudo Code:

* Output “Please enter value of first card:”
* Enter value of first card and store in Card1
* Output “Please enter value of second card:”
* Enter value of second card and store in Card1
* Output “Please enter operator +, -, \* or /:”
* Enter value of operator and store in Operator
* If Operator = +
  + Calculate Card1 + Card2 and store in Result
* Else if Operator = -
  + Calculate Card1 - Card2 and store in Result
* Else if Operator = \*
  + Calculate Card1 \* Card2 and store in Result
* Else
  + Calculate Card1 / Card2 and store in Result
* Output “Result:” and value of Result

## Solution 1.6 – Simple Loop

Pseudo Code:

* Output “Please enter value of first card:”
* Enter value of first card and store in Card1
* Set Count to 0
* Set Sum to 0
* Loop until Count = Card1
  + Output “Please enter value of next card:”
  + Enter value of next card and store in NextCard
  + Calculate Sum + NextCard and store in Sum
  + Calculate Count + 1 and store in Count
* Output “Sum:” and value of Sum

## Solution 1.7 – Loop with Exit

Pseudo Code:

* Set CardSuit to blank
* Set Sum to 0
* Loop until CardSuit not equal Heart
* Output “Please enter value and suit of next card:”
* Enter value of next card and store in NextCard
* Enter suit of next card and store in CardSuit
* Calculate Sum + NextCard and store in Sum
* Output “Sum:” and value of Sum

## Solution 1.8 – Loop with Exit and Case

Pseudo Code:

* Set CountClub to 0
* Set CountDiamond to 0
* Set CountHeart to 0
* Set CountSpade to 0
* Loop until CountClub = 5 or CountDiamond = 5 or CountHeart = 5 or CountSpade = 0
  + Output “Please enter suit of next card:”
  + Enter suit of next card and store in CardSuit
  + If CardSuit = Club
    - Calculate CountClub + 1 and store in CountClub
  + Else if CardSuit = Diamond
    - Calculate CountDiamond + 1 and store in CountDiamond
  + Else if CardSuit = Heart
    - Calculate CountHeart + 1 and store in CountHeart
  + Else
    - Calculate CountSpade + 1 and store in CountSpade
* Output “Count Club:” and value of CountClub
* Output “Count Diamond:” and value of CountDiamond
* Output “Count Heart:” and value of CountHeart

### Output “Count Spade:” and value of CountSpade