Software Engineering

Week 1 – Chapter 1

Evidence

*IQ Game Flowchart:*

*A diagram of a flowchart

Description automatically generated*

*This is the flowchart for the IQ Game where a user inputs a number lower than 10. If it is not lower, the program prints “Enter a number lower than 10” and restarts the program. If it is equal to 9, the program prints “You win!”, if it is lower than 9, it prints the number + 1 and “I win”. In both outcomes, the program asks the user if they want to play again. If the user types “J”, the program starts again, and if they type any other letter, the program ends.*

*At first, I wanted to make it so that when the person types “N”, the program ends, and when they type “J”, it starts again, and when the user types in any other letter the program prints “Wrong input.”. This is the original flowchart.*

*A diagram of a flowchart

Description automatically generated*

*However, I was advised to remove the “Invalid input” outcome and stick to the one where typing “N” or any other letter makes the program end, and only typing “J” makes it restart.*

*In the flowchart I made before this one, I made an if-condition that checks if the number is lower than 10. I was told that it was unnecessary because the condition N >= 10 already checks if the number is lower or higher, so I removed the unnecessary if-condition. This taught me how to make a program more efficient.*

*IQ Game Code:*

*A screen shot of a computer program

Description automatically generated*

*In the IQ Game code, I used the Boolean playAgain so that I can loop the while-cycle when the user types “J” when asked “Do you want to play again?”. I started with it being True, so the code is in action when running the program. The variable changes its value to False, when the user types “N” or any other letter, meaning they do not want to play again, so they get out of the loop and the program ends. When I presented my code, I was shown how to format it. That was the one thing I learned to achieve "clean code" and better readability.*

*Find the Leap Year Flowchart:*

*A diagram of a flowchart

Description automatically generated*

*This is the flowchart for the Find the leap year task where a user types a year, and the program checks different if-conditions to determine whether it’s a leap year or not. After every outcome, the program ends. When I showed this flowchart, I thought I can’t cross lines, so that’s why this flowchart looks strange. However, I learned that it’s okay and when lines cross in a flowchart, there should be a crossover bump, so that’s what I am going to do next time.*

*Find the Leap Year Code:*

*A computer screen shot of text

Description automatically generated*

*In this code I simply followed the steps of my flowchart, and it worked. The feedback I got was that I did well.*

*Here are the links to the assignments:*

IQ Game

Flowchart - [..\..\..\Fontys\Week 1\Chapter 1\IQ Game\IQ Game Flowchart.pdf](file:///C:\Fontys\Week%201\Chapter%201\IQ%20Game\IQ%20Game%20Flowchart.pdf)

Code - [..\..\..\Fontys\Week 1\Chapter 1\IQ Game\IQ Game.py](file:///C:\Fontys\Week%201\Chapter%201\IQ%20Game\IQ%20Game.py)

Find The Leap Year

Flowchart - [..\..\..\Fontys\Week 1\Chapter 1\Leap Year\Leap Year Flowchart.pdf](file:///C:\Fontys\Week%201\Chapter%201\Leap%20Year\Leap%20Year%20Flowchart.pdf)

Code - [..\..\..\Fontys\Week 1\Chapter 1\Leap Year\Find the leap year.py](file:///C:\Fontys\Week%201\Chapter%201\Leap%20Year\Find%20the%20leap%20year.py)