Structure English vs. Pseudocode vs. Executable Code

Structured English is very similar to Pseudo code, but it tends not to use so many mathematical symbols. Often people start with Structured English, convert it to Pseudo Code and then write Executable Code.

|  |  |  |
| --- | --- | --- |
| **Structured English** | **Pseudo Code** | **Executable Code** |
| **READ** name  **IF** name **EQUAL** "Harry" **THEN**  **WRITE** "Why don't you marry Pippa?"  **ELSE**  **WRITE** "Are you Royal enough?"  **END IF** | **INPUT** name  **IF** name **= =** "Harry" **THEN**  **OUTPUT** "Why don't you marry Pippa?"  **ELSE**  **OUTPUT** "Are you Royal enough?"  **END IF** | **Sub Main()**  **dim** name as string  **name =** console.readline()  **if** name **=** "Harry" **then**  **console.writeline**("Why don't you marry Pippa?")  **else**  **console.writeline**("Are you Royal enough?")  **End if**  **End Sub** |

**FAQs**

**Q. What are the rules of Structured English?**

A. There are no clear rules, but you must use **a small subset of the English language** and **a few simple conventions**; a sequence **instruction** starts with a **verb** and an **assignment** uses the term **SET**.

**Q. What are the rules when writing pseudocode?**

A. There are no set rules, but the code should provide clear descriptions of the algorithms being outlined; you should use **relational**, **arithmetic**, **Boolean operators**, **actual variable names**, **equivalent =** should be replacedwith **==** and the **assignment =** should be replaced with **🡨** or **:=**.

1. **What is the difference between pseudocode and a programming language?**
2. Pseudocode cannot create executable code, whilst programming languages can.

Questions:

1. Write the pseudocode for the given Structure English: **[Grade D]**

|  |  |
| --- | --- |
| **Structured English** | **Pseudocode** |
| **Set** running total and running count to 0(zero)  **Read** in the first number  **Repeat** the following until the number is 0  **Keep** a count of the running total  **Keep** a count of how many numbers read so far  **Read** in the next number  **Set** average to total **divided** by how many numbers  **Print** average | Running\_total <= 0  Running\_count <= 0  no <= USERINPUT  Repeat  Running\_total <= Running\_total + no  Running\_count <= Running\_count + 1  Until no == 0  Average <= Running total / Running\_count  Output Average |

1. For the following pseudocode write a VB.NET equivalent: **[Grade E]**

|  |  |
| --- | --- |
| **Pseudocode** | **Executable code (VB.NET)** |
| Input Number1  Input Number2  Sum 🡨 Number1 + Number2  Average 🡨 Sum/2  Output Sum, Average | Dim num1, num2, sum, average As Single  Num1 = Console.readline()  Num2 = Console.readline()  Sum = num1 +num2  Average = sum /2  CWL(Sum + Average) |

1. Write the Structured English, Pseudocode and Executable Code for the following problem: **[Grade C]**

**A program for a climate control unit that opens windows if the temperature is above 25 and activates the heating if the temperature is below 10.**

|  |  |  |
| --- | --- | --- |
| **Structured English** | **Pseudocode** | **Executable Code** |
| READ temp  IF temp MORE THAN 25 THEN  SET heating AS TRUE  ELSE IF temp LESS THAN 10 THEN  SET windows to open  ELSE  SET heating AS FALSE  SET windows to closed  END IF | Temp = read temperature  Case temp Of  >25 : activate heating  <10 : Opens windows  End Case | Dim temp As Single  Temp = Readtemp  Select Case temp  Case Is < 10  activate heating  Case Is > 25  Opens Windows  End Select |