**Programming Solutions**

**Programming Concept: (e.g. Nested IFs)**

|  |  |  |  |
| --- | --- | --- | --- |
| **Q / TASK** | **Code** | **Console Output** | **Comments/Problems** |
| 1 | Global: Const NoOfTraps = 2  Const NSDistance = 5  Const WEDistance = 7  Local: All other variable (in Sub and Function)  e.g. Dim Cavern(NSDistance, WEDistance) As Char  Dim Choice As Integer  Dim FlaskPosition As CellReference  Dim MonsterAwake As Boolean  Dim MonsterPosition As CellReference  Dim PlayerPosition As CellReference  Dim TrapPositions(NoOfTraps) As CellReference  Dim Count1 As Integer  Dim Count2 As Integer  Dim Position As CellReference |  |  |
| 2 | Module Module1  Dim globalvar As String = "hi"  Sub Main()  Dim localvar As String = "there"  Console.WriteLine(globalvar + localvar)  Another()  Console.ReadLine()  End Sub  Sub Another()  Console.WriteLine(globalvar)  End Sub  End Module |  |  |
| 3 | |  |  |  | | --- | --- | --- | | Scope of variable | Accessibility: where to read/ written to, Variable Lifetime | | | Scope of Procedure | Where a procedure can be called (by the context you are allowed) | | | What does it called | Local variable | Global variable | | Term | Private | Public | | Where | Inside  Module | Inside  Module | | Where  (How to identify it) | Inside  Sub  Function | Outside  Sub  Function | | definition | Only used in block defined | Can be used Anywhere in or outside of a project | | Example | Sub Main()  Dim local1 as string  End Sub  Function hi()  Dim local2 as string  End Function | Dim global as string  Sub Main()  End Sub  Function hi()  End Function | | Benefit | = Save the quota of global variable  = Easier to debug, understand the issue when multiple task is running together  = Easier to recognize as you declared in  = Save actual memory space  = Protection for variable doesn’t change by other program | = Use/ Access/ Visible the variable everywhere in the code  = Provide data sharing  = Don’t have to create when needed to use: faster | | Drawback | = Cant share the variable to other Function/Procedure unless ByVal or ByRef  ( Have to pass by parameters)  = Data lost when the function is terminated  = Change of the local variable is not visible by another function | = Hard to debug (use/mention many times)  = Anyone can change the global variable  = Data lost when the program is terminated  = Out of memory issue | | Differences | = Local variable only spawn when the function/ procedure is called  = Temporarily spawned in RAM stack spawn  Stored on the stack | = Global variable is spawn at the beginning of the program  = Allocate in user RAM space  = Stored on a fixed location decide by compiler | |  |  |
| 4 exten |  |  |  |
|  |  |  |  |
|  |  |  |  |

Note: Please make sure you save this document as ‘**if\_solution\_yourname.doc’**