e-Journal of Business Education & Scholarship of Teaching
Vol. 7, No. 2, 2013, pp: 46-58.
"http://www.ejbest.org"

Instructional Note

A String Search Marketing Application Using Visual Programming

Jerry M. Chin *
Computer Information Systems Department
Missouri State University
Springfield, MO 65897, USA
jerrychin@MissouriState.edu

Mary H. Chin
Marketing Department
Missouri State University
Springfield, MO 65897, USA

Cathryn Van Landuyt
Computer Information Systems Department
Missouri State University
Springfield, MO 65897, USA

ABSTRACT

This paper demonstrates the use of programing software that provides the student programmer visual cues to construct the code to a student programming assignment. This method does not disregard or minimize the syntax or required logical constructs. The student can concentrate more on the logic and less on the language itself.

Keywords: Graphical programming; introductory programming; string search application.

JEL Classification: A22, C02
PsycINFO Classification: 3530
FoR Codes: 0803; 1302

- Corresponding author, 417-836-4131, Fax 417-836-6907

Introduction - More Than Just A Problem

Too often the first step after the professor hands out the next programming assignment is the rush to the computer keyboard. There is no quiet contemplation of the problem identity, requirements of the tasks, data structures or, most important of all, the logic. The student should realize that basic program flow can be sequential, branching and looping. The trick in programming is the judicious choice of the flow changes. Assuming there is a flowchart or pseudo-code, the next step is the actual creation of the code for the target problem.

The PWCT approach to programming is to relieve the beginning student from some of the "point & click" procedures of most computer languages. Through the use of a familiar diagrammatic visual presentation the burden of syntax is lessened and the emphasis can be turned more to the teaching the development of logic. By the use of diagrams, the student fills in textboxes to provide the necessary information of the instruction or data construct under consideration. In addition, the prompting of the "fill-in" diagrams provides some insight to the semantics - meaning to the instruction or flow construct.

Learning Objectives

A student working through this problem will be exposed to a problem not ordinarily encountered in t he business college curriculum. The first course of logic usually originates from mathematics or philosophy. From a programming course standpoint, some student objectives are:

- Basic string functions; strings are examined as a simple data structure. Individual string characters can be isolated, examined, and processed.
- While loop; the implementation of the loop demonstrates an example how to enter the loop and how to exit a loop.
- Code modularity; define a function, pass parameters to the function and return to the logic flow.
- String utility functions; We replace a substring with another substring resulting in a new string.
- High level language expectations; By learning one high level language, the student gains insight of what to expect from another high level language. The student sees If-Then, loops, and function calls
- Problem analysis; a close inspection of the problem discloses that there are three basic types of substring families. Once the family has been determined, the actual substring can be extracted and processed.
- Syntax and semantics; instruction form and meaning. What does a proper instruction look like and what does it do?

A String Search Application Using Visual Programming

Programming Without Coding Technology (PWCT) is a free software package that provides a GUI which allows the student to construct a program by using an interface with pop-up boxes and templates to reduce the need to know the exact syntax. The semantics and knowing when to use instructions is part of the logic or "coding" process.

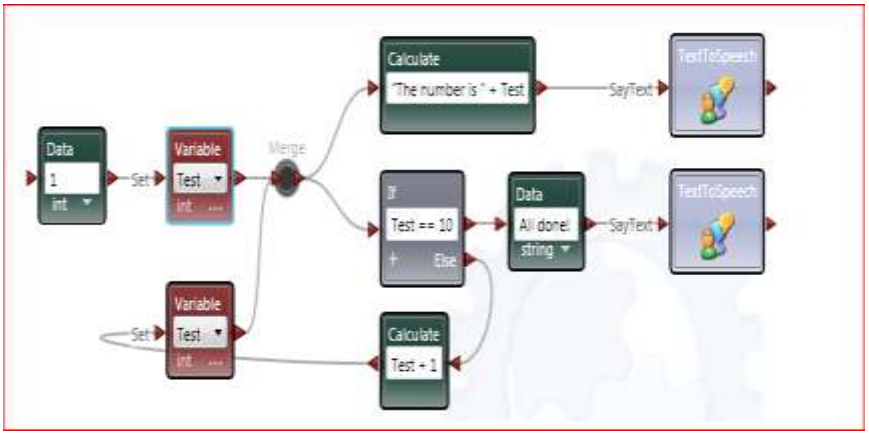
The student still is required to have a logical overview of the code. For the problem presented in this paper, the string functions provided were sufficient. The GUI provides a structured way to control the levels of code such as code inserted within a While-Loop or an If-Then statement.

Visual Programming

There are a number of software packages that are visual programming languages. We list three, two of which are free downloads.

Microsoft VPL (VPL, 2011) is a graphical programming package that is part of the Microsoft Robotics Developer Studio. This application is used to guide and operate robots. The philosophy uses the graphical dataflow-based programming model, where the components initiate action once all the signals or messages arrive rather than a sequence of commands executed in sequence. The language purportedly targets the beginning programmer. The following shows the variable being bumped by one until it reaches ten.

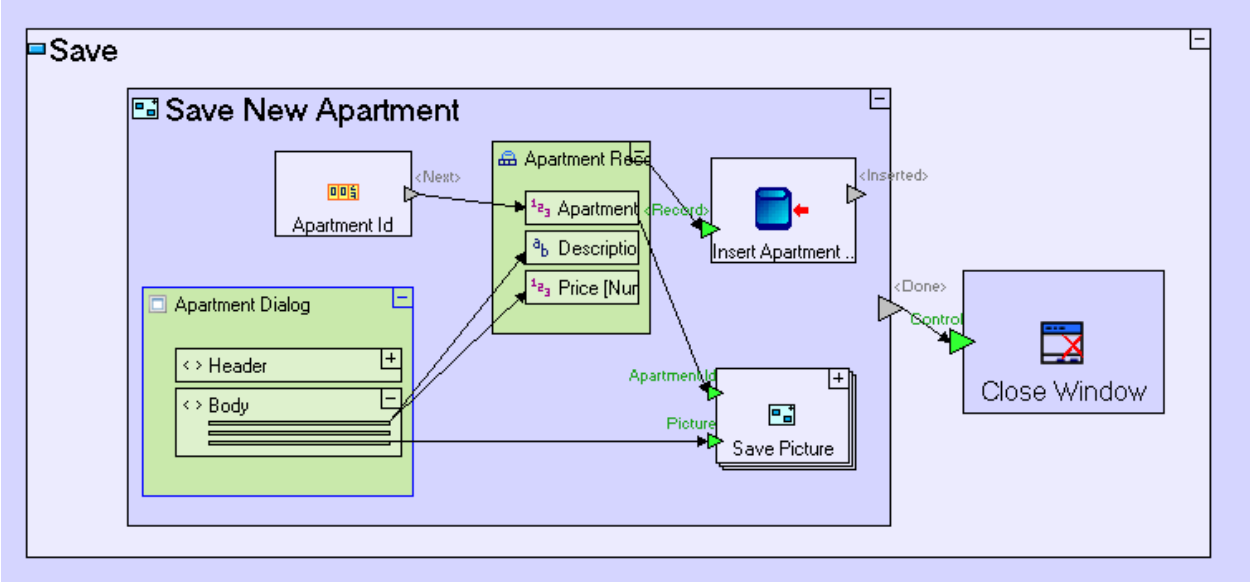
Figure 1:
Graphical Data Flow Diagram- Microsoft VPL



The programmer chooses from a list of services or functions which are dragged and dropped on a diagram window. By opening the graphical representations the programmer is prompted to supply the name of values or variables when needed. Connections are easily made by linking output nodes to input nodes. As can be seen in the example, inputs can be merged when appropriate.

Tersus (Tersus, 2011) is a language based on drawing graphical diagrams rather than writing statements. The following picture displays an example.

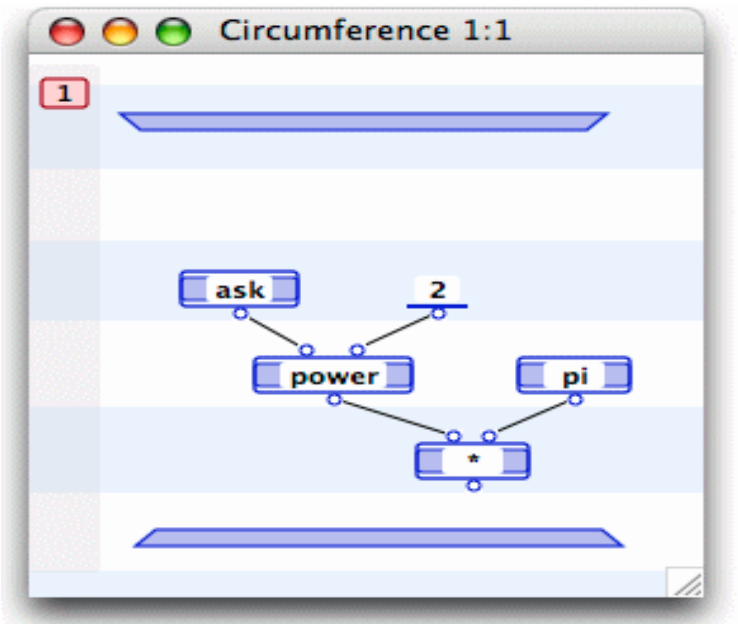
Figure 2:
Graphical Data Flow Diagram- Tersus



In the diagram, clicking on a component enlarges the graphic so that other components can be inserted. In this way, a simple flow into a component can be controlled as other compnents to be processed are layered within a component.

Marten is based on Prograph (Steinman & Carver, 1995) and currently available at the Web site, http://andescotia.com/support/ . Marten's approach is the insertion of graphics which are connected via data links. Entry terminals set values and output roots to pass on results. A sample program for the area of a circle is depicted in the following figure. The "power" block has two entry roots and one output root.

Figure 3:
Graphical Data Flow Diagram- Marten



The Problem

This student problem is based upon the truth-functional form algorithm found in the study of first order logic. FOL is a study of sentences and the logical consequences when these sentences are considered in a specific framework. In particular, the algorithm boils down to a string search and replace problem (Barwise and Etchemendy, 2003, p. 261).

Eastern Imports has a legacy distribution system where internal records show the manufacturing plant, item and destination. Because of the competitive marketing conditions, customers have asked for more confidentiality of their business purchases. Eastern has decided to code their invoices when external intermediary shippers are involved. Plants and destination address are all that are revealed in Table 1.

Table 1:
Plants and Destination Addresses

Plant	Distribution Warehouse
D = Dallas	L = Los Angeles
A = Atlanta	N = New York City
DxLx AxNx	Ship x from Dallas to Los Angeles Ship x from Atlanta to NYC
Ny	Pick up y from New York City
Lw	Pick up w from Los Angeles
Dx(note) Ay(note)	Dallas special order for x Atlanta special order for y

Logic of the Search

Upon inspection, the problem analysis reveals three possible forms of substrings, listed in the table above. We categorize the three forms in Table 2:

Table 2:
Three Forms

F1: e.g. Lx, Ny
F2: e.g. DxLx, AyNy
F3: e.g. Ay(5% discount by Dec 24)

The rationale for Table 3 follows. Let S_i be the character string at the i^{th} position in string S. If S_i is a substring of "DALN", then we have we can search the string to determine if the form is F1, F2, or F3. Otherwise, bump in the index of S to i+1 and

look at S_{i+1} . If S_i = "A" or "D", then we have F2 or F3. Check S_{i+2}. If S_{i+2} = "L" or "N", the form is F2 and its length is 4. If S_{i+2} = "("then the form is F3. At this point, we must determine j such that S_j = ")". The length of the F3 substring is j-1+1. If S_{i+2} is not a substring of "LN(" then by default the form is F1 and its length is 2.

Table 3:
Character Strings

S_i	S_{i+1}	S_{i+2}	S_{i+3}	Form	Length
A	x	N	x	F2	4
D	X	(		F3, where S_j =)"	j-i+1
L	y			F1	2

A message in the system might look like:

S = "DxAx+ Ay(5% discount by Dec 24)+AyNy+DxAx+Ny+Ay(5%discount by Dec24)+Ld"

The security decided upon is to transform the string above to a message with the keys for the particular message kept separarely. The algorithm decided upon is to search the string beginning on the left and substituting like substrings by a single substitution character. For example, suppose that the set of substitution characters are theletters {B,C,E,F,G,H}. Then the transformed string S has the form S = "B+C+F+B+G+C+H".

Note that B and C are repeated twice, indicating a multiple occurrences of two substrings, "DxAx" and "Ay(5% discount by Dec 24)"which were replaced by "B" and "C", respectively. The string S has been transformed into a simpler string while preserving the general structure of S.

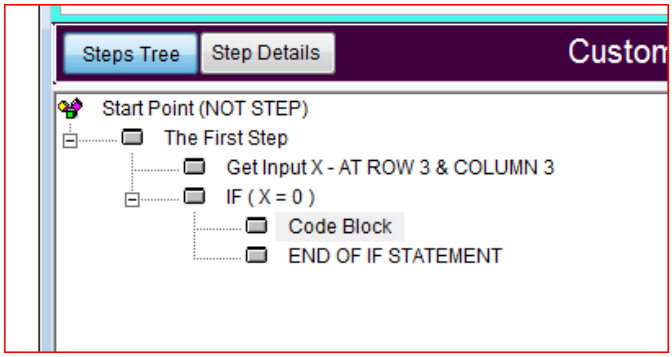
The Code

The user interface for PWCT prompts the user so that the actual syntax is not as critical as in traditional software packages. Choosing one option produces a list of choices to control the program flow of logic. For example, if we choose an IF Statement, the following window pops up.

Figure 4:
Screen shot of the IF Statement

Three choice boxes can be checked to produce an IF-Then, an IF-Then-ELSEIF, or an IF-Then-Else control structure. If the text condition is X = 0 and we check the first checkbox, a template for IF-Then code is inserted. By clicking on the Code Block, insures that the next code will be inserted at the right place inside the IF-Then block. This is an example of the aforementioned fill-in diagram. Syntax is not a problem.

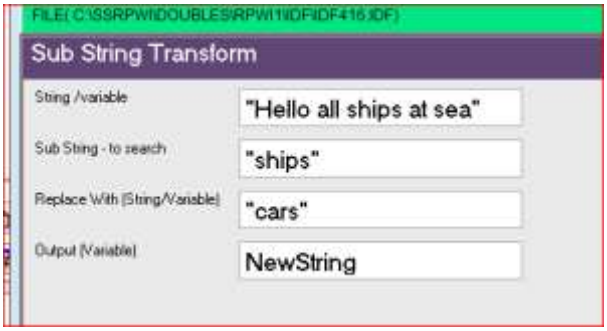
Figure 5:
Screen shot of the IF Statement Inserted into code



Sub String Transform Function

This string function finds a string, Si, within a string S, and replaces all occurrences of Si with another string L. For example, if S = "Hello all ships at sea", Si = "ships" and L ="cars", then NewString ="Hello all cars at sea". The prompting diagram looks like:

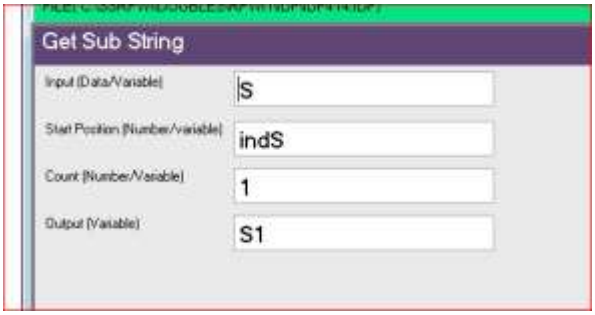
Figure 6:
Screen shot of the Sub String Transformation



Sub String Function

There will be a need to examine a substring given a starting position i and length of the string. In particular, if the count (length) is one, the Get Sub String function below can be used to examine a single character of String S starting at position indS. This is shown in the following diagram;

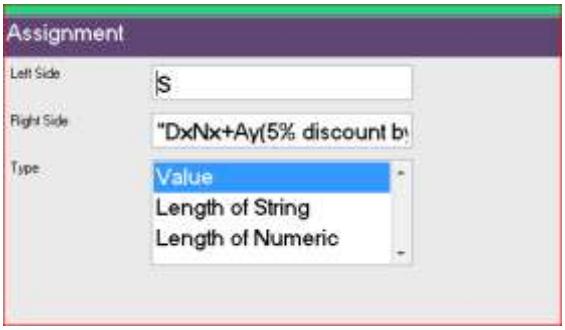
Figure 7:
Screen shot of the Get Sub String



Assignment to a Variable

The assignment function requires the left-hand side for the variable and its right-hand side value. For example, if x is the left-hand side and zero is the right-hand side, the syntax is x = 0. The semantics is self-evident. Below is an example of the String S being assigned with a right side string value, "DxNx+Ay(5%....".

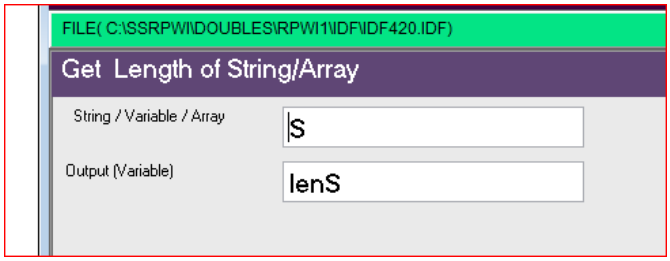
Figure 8:
Screen shot of the Assigned A Variable



Length of a String

Similarly, the length of a string can be captured by a two-value pop-up box as seen below. If S ="123456" then lenS = 6.

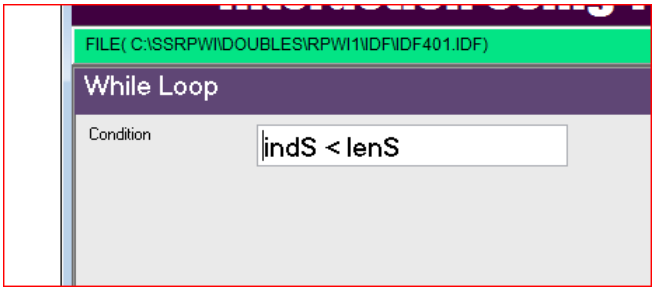
Figure 9:
Screen shot of the Get Length of A



The While Loop

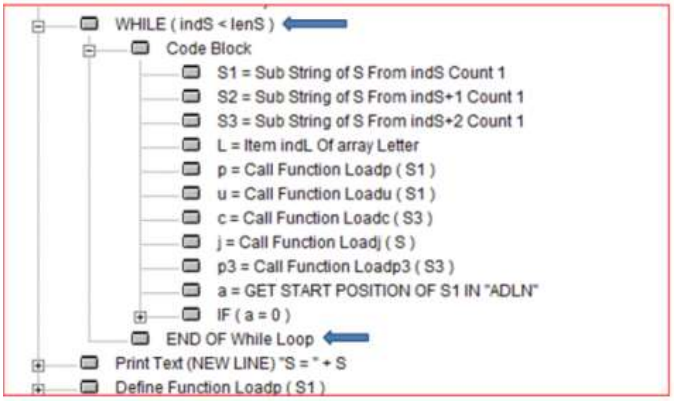
The While loop is a control structure which checks the condition upon entrance into the loop.

Figure 10:
Screen shot for setting the condition in a While Loop



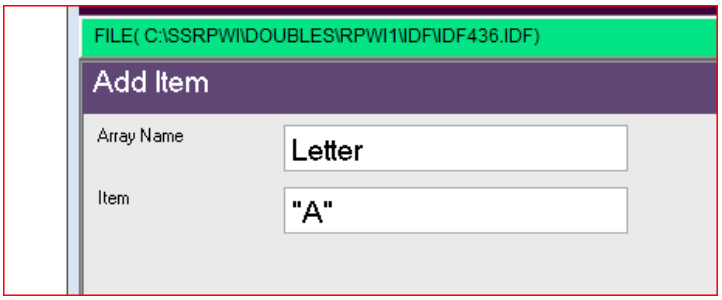
In the problem at hand, the While loop compares the position index to the length of the string S. The resulting While loop is shown is the figure below with the beginning and end of the construct marked with arrows. The Code Block is a marker which the user can click so that subsequent statements can be inserted at the appropriate level.

Figure 11:
Screen shot for the Code of a While Loop



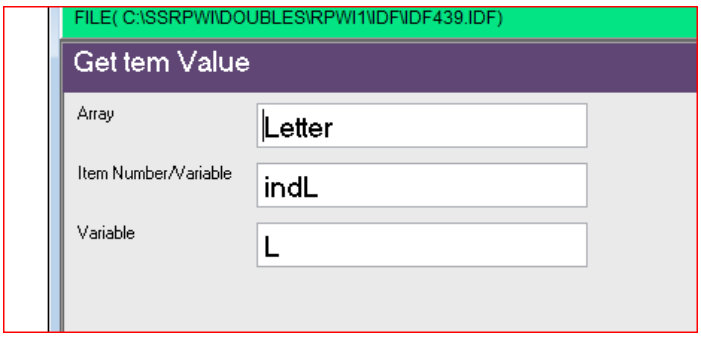
Creating an Array
After an array is created, items are inserted using the Add Item operation. In the example below, the string "A" is inserted into an array called Letter.

Figure 12:
Screen shot for adding an item to an Array



For example, to access an item in an array called Letter, array index indL and assignment variable L are designated. This operation is shown below.

Figure 13:
Screen shot for retrieving an item from an Array

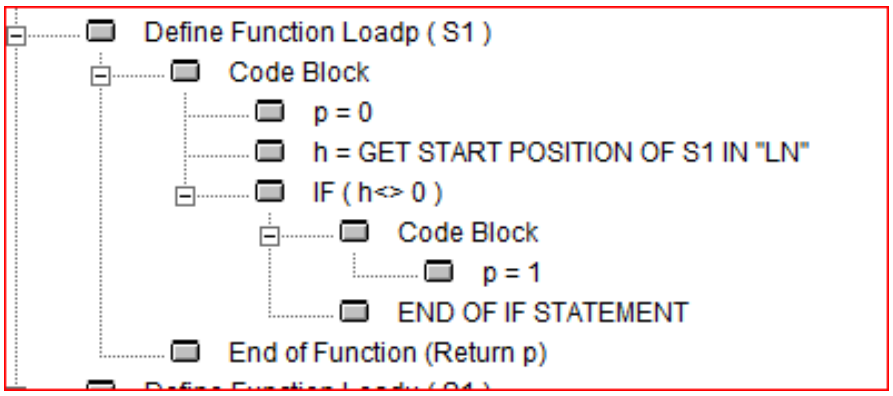


Functions

As expected, code modularity is supported by defining a function, passing parameters and calling the function. In the following diagrams, five functions are defined in this paper's example program code.

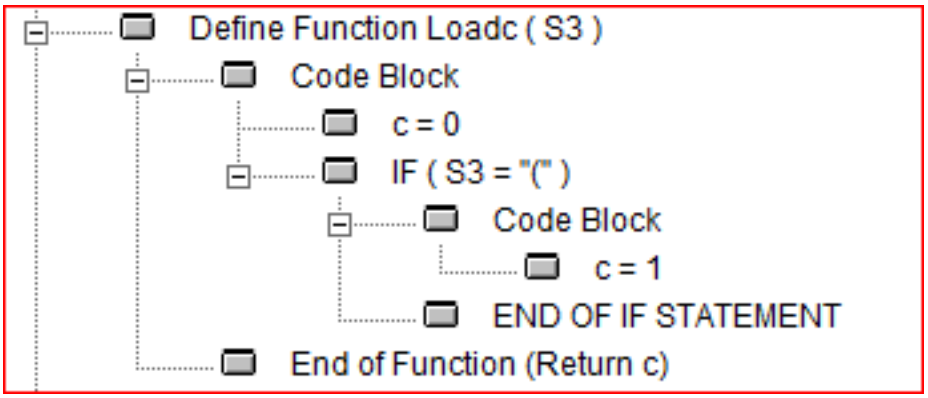
Does the one character string S1 equal to a "L" or "N"? The Loadp function below reports the index of a string "LN" where the string S1 matches. The index is preset to zero. If S1 is equal to a "L" or "N", then the index will be not zero. If not zero, then p is assigned a value of 1.

Figure 14:
Screen shot for Function Loadp



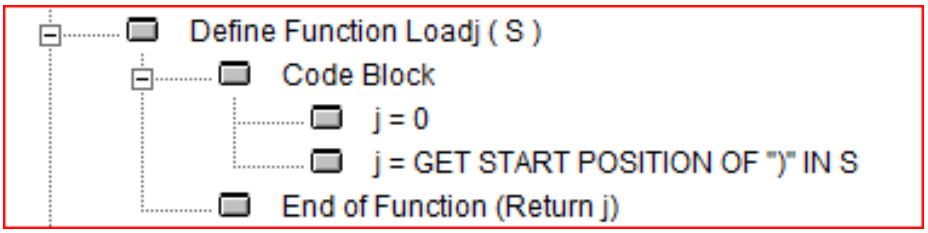
Similarly, a function checks for the occurrence of "A" or "D".Loadc answers the question if S3 is equal to the character ")".

Figure 15:
Screen shot for Function Loadc



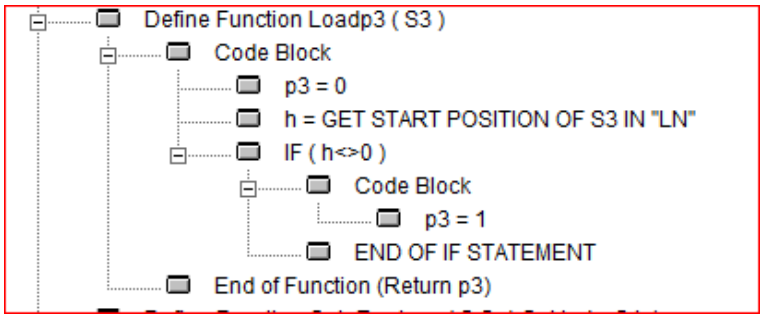
Loadj reports the first occurrence at position j of the character ")" .

Figure 16:
Screen shot for Function Loadj



Loadp3 determines if the string S3 is equal to a "P" or a "Q".

Figure 17:
Screen shot for Function Loadp3



GetnReplace first finds in S a substring W with length Count starting at position indexS in S. The next determines cnt which is where W first occurs in S. A While-loop based on cnt is entered. In this loop W is replaced with the string L, resulting in a new altered S. Still inside the loop, the next occurrence in the new S is determined and a new value of cnt is also

determined. When cnt is zero, W does not appear in the new S anymore. In this way, the original S has been transformed by replacing every occurrence of W by L.

Figure 18:
Screen shot for Function GetnReplace

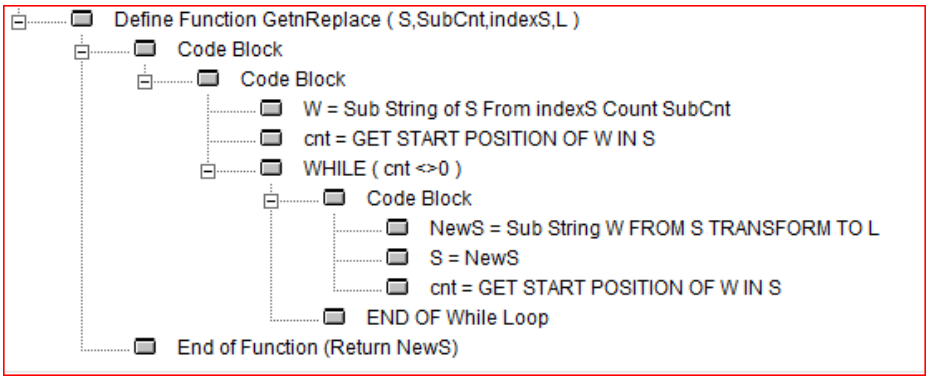
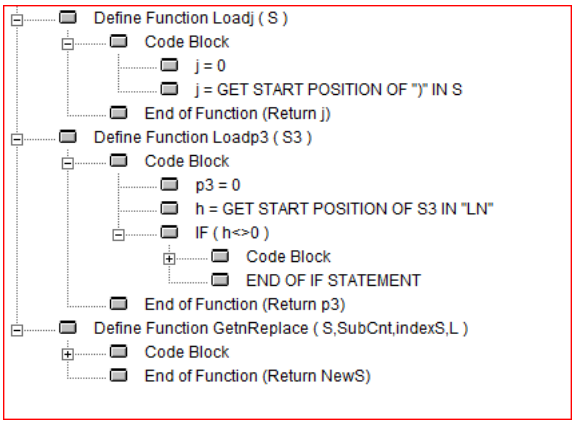


Figure 19:
Screen shot for Various functions displayed in code



Teaching Points

The PWCT software still requires that the user have a familiarity of computer structures such as the While-Loop and the If-Then. This software shows the user to some extent how to formulate the code. Moreover, there is the expectation of procedures being defined and called along with accompanying parameter lists and returned values. The when-and-where is determined by the solution logic. Arrays were used to store and provide values as the logic demanded. Syntax errors are decreased since the software prompts the user. The user provided the variables and the logic to solve the problem. For the beginner, this system lets the user concentrate on the problem and the programming logic.

Conclusion

The discussion above was brief and the illustrated code snippets provided a flavor for the PWCT software. In practice, the experienced programmer knows what to expect from a language. Familiar logic constructs such as looping or data structures such as arrays were found to be in the programmer's tool box. The strength of the PWCT is in the prompting diagrams since the syntax issue is reduced. The logic and analysis of the program solution are still created, developed and judged by the student.

References

Programming Without Code Technology,PWCT. [Computer Software]Version1.7 (Sharp) 2010.09.15. Available at http://doublesvsoop.sourceforge.net/

Tersus. [Computer Software] Available at http://www.tersus.com/

Microsoft VPL [Computer Software] Available at http://msdn.microsoft.com/en-us/library/bb905471.aspx

Barwise, J. & Etchemendy, J. (2003). *Language Proof and Logic.* Stanford, CA : CSLI Publications.

Steinman S. & Carver, K. (1995). *Visual programming with Prograph CPX*. Greenwich, CT: Manning Publications.

Marten. (1991). *Aiming higher (Commissioned Report No 1).* Melbourne, Australia. Available at http://andescotia.com/support/ Business/Higher Education Round Table [Computer Software] Available at http://andescotia.com/support/