Program Development

1. Understand the problem. Identify the inputs and the outputs.
2. Find an algorithm or several algorithms. Select the best algorithm to use.
3. Put the algorithm on paper. (pseudo code or graphical solution – flowchart)
4. Code the flowchart in an appropriate language.
5. Translate the instructions and debug your program, until error free.
6. Run the program with tested data.
7. Document the program with useful comments and program run.

Flowcharting Symbols

1. Start/End Symbol - oval

Start

End

1. I/O Symbol – parallelogram

input x, y

output 2 \* x + y

1. Processing symbol - rectangle

doubleX = x \* 2

doubleY = 2 \*y

1. Decision Making Symbol – Diamond

T

doubleY > 50

F

1. Continuation Symbol - circle

A A

**Flowcharting Logic Patterns**

1. **Simple Sequence**

input x,y

prod = x \* y / 3

output x, y, prod

1. **Decision Making Logic Pattern** – where you test some condition and if **TRUE**, do one or many instructions but if **FALSE** DO another 0, 1, or many instructions.

F T

x >= y

y = y + 10 Example 1

x = x – 3

F T

x >= y

input c, d y = y + 10

x = 2\*c x = x – 3 Example 2

y = c + d

1. **Repetition Logic Pattern** (with **pre-test** ) - Uses the **WHILE** statement!

This allows for repeating some code (called the body of the loop) while the condition is **True**. When the condition is finally **False**, it leaves the loop

. . .

input ID

F

ID != -999 D input an ID value

T If ID == -999 leave loop

. . . (input some values) If ID != -999, continue loop

Processing values

body of loop . . . (process values) Output answers

Input a new ID value

. . . outpu answers

. . (output answers) Repeat the loop

Input ID

D Continue at D, if ID != -999

(continue flowchart after loop)

1. **Repetition Logic Pattern** (with **post-test)** - Uses the **WHILE** statement using **WHILE True:** clause. This allows for repeating some code until a condition allows control to leave the repetition, at the **bottom** of the loop. (While the condition is TRUE, it will do the body of the loop, and decide at the bottom of the loop if it should repeat the body of the loop or not! The test to repeat the loop is at the **bottom** of the loop and it uses an **IF** statement to finally break out of the loop.

. . .

True

T Input values

. . . (input some values and calculate with values

body of loop output the answer

. . . (process values) if answer <= 500 loop back

If answer > 500, leave loop

. . . outpu answers

. . (output answer)

(Note: test to leave loop

F answer > 500 is at the bottom of loop.)

T

. . .