How are Algorithms Put Together?

• Sequential instructions do them in the order given
• Conditional instructions do them if a condition is true
• Iterative instructions do them while a condition is true

Steps in Program Development

1. Analyze Problem
   - Black box or functional design.
   - Example problem statement:
     Write a program that gets the height, width, and depth of a box from the user and reports the volume of the box. A program that gets the height, width, and depth of a box from the user and reports the volume of the box.
   - Input: Height, Width, Depth
   - Given value: None
   - Processing: Calculate the volume
   - Output: Volume of the box

2. Outline Solution
   - Major steps/functions/boxes
   - Iterative process: Go back to step 1 as needed!
   - Design strategies:
     - Top-down
     - Bottom-up
     - Inside-out

3. Pseudocode the Algorithm
   - Pseudo or apparent rather than actual
   - Will develop later

4. Test Algorithm

5. Program Solution

6. Run and test the program

7. Document and maintain the program

Steps in Program Development
Step 4: Test the Pseudocode

- Test cases
- Boundary cases
- Normal cases
- Correctness (rather than efficiency)

Testing is manual.

Return to step 1, 2, or 3 to fix problems.

Step 5: Code

- Pseudocode “close” to code, but more abstract
- Computer instructions
- Programmer time is expensive
- Machine time is cheap
- Many details involved in coding in C++

Step 6: Run and Test

- Rewarding step
- Debugging could force return to previous steps (s)

Step 7: Document and Maintain

- Documentation – essential
- A program is a means of communicating to other programmers, designers, and programmers.
- Maintenance: 70-90% of software cost
- For designers
- For programmers
- For the user

Step 4: Test the Pseudocode

- You are master, computer is slave
- Simple
- Specific
- Detailed
- Computer instructions

Pseudocode

- contains keywords (words that have special meaning)
- contains variables (stores a single value, which can change)
- contains operators (operation is to be performed)
- contains punctuation (quotation marks: used to indicate the beginning and end of a string of text, commas: used to separate items in a list, dots: used to separate the items in a list, etc.)
- English-like, but unambiguous
- flexible (no rigid syntax rules)
Keywords – Special Meaning

Get Else If
Print End If
And Repeat
Or End Repeat
If While
Else End While

Variables
• Addressed box model
• Assign/Write: Store a value / destructive
• Use/Read: Retrieve value / non-destructive, current value remains in the location
• Value is copied for us to use

Operators
• same as in math: + - * / ()
• Example:
  count = count + 1
• Example: count = count + 4
• Example:
  count = count + 4

Exam Average = (score1 + score2 + score3) / 3
Average = sum / count
Result = 95 * 63 / 12
Sum = 5 + 4 + 19

Basic Calculations

Form: > variable = > math expression

Exam Average = (score1 + score2 + score3) / 3
Average = sum / count
Result = 95 * 63 / 12
Sum = 5 + 4 + 19

More Operator Examples

Descriptive Variable: count, length, width
Easier to maintain
More readable to you and others
Name is vital, important
Different naming conventions
• Single word
• Unique
• Descriptive
We name variables

More About Variables

Non-destructive: current value remains in the location
Use/Read: Retrieve value

Value is copied for us to use

Whenever used, there is gone
Destructive
Assign/Write: Store a value
Addressed box model
Displaying messages and values

form: Print "<message>
form: Print <variable>
form: Print "<message>", <variable>

Examples:
Print "Hello!"
(display the message "Hello" on the screen)
Print cost
(display the current value of variable cost on the screen)
Print "The average is ", average

Getting input
form: Get <variable>

Examples:
Print "Enter the price: "
Get price
(get the value for variable price from the user's keyboard)

Pseudocode Example
Write a program to calculate the volume of a box, using dimensions provided by the user.

Print "Enter height, width, and depth of box:"
Get height
Get width
Get depth

volume = height * width * depth
Print "Volume is: ", volume

Selection Instruction
form: If <comparison>
<statements executed if true>
Else
<statements executed if false>
End If

comparisons use:
<, >, <=, >=, !=, ==, And, Or

Example
If age < 2
Print "Baby"
Else
If age >= 2 AND age < 11
Print "Kid"
Else
If age >= 11 AND age < 19
Print "Teenager"
Else
Print "Cool"
End If
End If
End If

Displying messages and values
form: Print <variable>
form: Print "<variable>
form: Print "<variable>", <variable>
form: Print "message", <variable>
form: Print "message: "<variable>
form: Get <variable>

Examples:
Get price
(print the value for variable price from the user's keyboard)
Step 1: Analyze the problem

Example

Write a program that gets an employee's information from a user and calculates their pay for the week. The input includes the employee total hours worked this week and their hourly wage. The employee is to be paid their basic wage for the first 40 hours worked and time-and-a-half for all hours above 40. Output the regular pay, overtime pay (if present), total pay. If the employee worked 40 hours or less, do not output any information about overtime pay.

Step 1: Analyze the problem

• Inputs:
  - hours worked, pay rate
• Given values:
  - over 40 hours is overtime
  - overtime is 1.5 of regular pay
• Calculations:
  - regular pay, overtime pay, total pay
• Outputs:
  - regular pay, overtime pay (if present), total pay

Step 2: Outline solution

• Get hours worked, and pay rate from a user
• If there were overtime, print the overtime pay.
  - Else If selection instruction:
    - if statements executed if true
    - Else If selection instruction:
      - if statements executed if true
  - End If
• Else:
  - Print regular pay.
  - Else If selection instruction:
    - if statements executed if true
    - Else If selection instruction:
      - if statements executed if true
  - End If

Example

End If
Print "Cool"
Else
Print "Geezer"
Else If age > 30
Print "Geezer"
Else if age >= 11 AND age < 19
Print "Teenager"
Else if age >= 2 AND age < 11
Print "Kid"
Else if age > 2
Print "Baby"
End If
Step 3: Algorithm in Pseudocode

1. Print "Enter hours worked: 
2. Get hours 
3. Print "Enter rate of pay: 
4. Get payRate 
5. If hours > 40
6. regularPay = 40 * payRate 
7. overtimePay = \((\text{hours} - 40) \times \text{payRate} \times 1.5\)
8. Else 
9. regularPay = \text{hours} \times \text{payRate} 
10. overtimePay = 0
11. End If 
12. totalPay = regularPay + overtimePay 
13. Print "Regular Pay: ", regularPay 
14. If overtimePay != 0
15. Print "Overtime Pay: ", overtimePay 
16. End If 
17. Print "Total for this week: ", totalPay

Step 4: Test algorithm

- Provide some sample inputs and follow the algorithm exactly step by step, find out the outputs
- Decide which instructions need to be done over and over - Put these inside the loop
- Decide which instructions need to be done once - Put these outside the loop
- Repeat - Iterative Instruction form: Repeat <N> times <statements to be repeated>
- End Repeat
- While form: While <comparison> <statements to be repeated if true>
- End While

Repetition Observations

- Decide which instructions need to be done only once – Put these outside the loop
- Decide which instructions need to be done over and over – Put these inside the loop

Accumulator in Loop

- Increment in loop
- Assign initial value before loop
- Total = 0
- End

Step 3: Algorithm in Pseudocode
Write an algorithm to ask the user to input 10 numbers. The program will then return the sum to the screen.

```
total = 0
Repeat 10 times
  Print "Input a number"
  Get value
  total = total + value
End Repeat
Print "Total sum of 10 numbers: ", total
```

### Counter – Special Kind of Accumulator

- Initialize to 0 or 1 typically
- Increment by 1 in loop (keeps track of how many times through the loop)

### Examples

- Write a program to output the squares of the numbers 1 through 100
- Write a program to output the sum of a list of positive numbers entered by the user
- Write a program to output the sum of a list of numbers entered by the user
- Write a program to output the sum of a list of numbers entered by the user

Visual Studio 2005

- [http://www.cs.usu.edu/elms.htm](http://www.cs.usu.edu/elms.htm)
- Free download to students currently in a CS class
- Not for commercial use

Visual C++ Express Edition

- [http://www.microsoft.com/express/](http://www.microsoft.com/express/)
- Free to all
- Not for commercial use
- Free download to students currently in a CS class

```c
// Repeat 10 times
End Repeat
```