Algorithms

CSCI-101
What is an algorithm?

• There are many definitions, but here are the essentials:
• It is a step-by-step description of what needs to be done
• We need the detailed steps:
  – so we can clearly see what needs to be done
  – to force us to look at the details before we try to code a solution
Algorithms and Programs

• We already told you that later in the semester, you will be writing computer programs

• What is the relationship between an algorithm and a program?
Algorithms and Programs

• What is the relationship between an algorithm and a program?
• If you do a good job developing your algorithm, you should be able to hand the algorithm to anyone and ask that person to write the code – regardless of the language
• You could implement your algorithm in C, C++, Java, Python, ... it doesn’t matter! The algorithm solves the same problem, regardless of the implementation language
Steps in Your Algorithm

• Must be very specific
• Must involve simple operations
• Operations may be simple
  – Do <this> operation
• Operations may consist of arithmetic or logic instructions
  – Add these numbers
  – Calculate the square root
  – Only do this operation under certain conditions
  – Do this operation a designated number of times
Math Operations

• In your algorithms, express as you would in your math class by using parenthesis to clarify the order or operations

• Examples:
  – Calculate area = PI * (r^2)
  – Calculate avg = (a + b + c) /3
Logic Operations

• In your algorithms, express logical expressions in terms of yes/no or true/false (booleans)

• Examples:
  – If area is > 20
  – If avg = max
  – If color is not blue

• If you have a compound boolean, use parenthesis to clarify:
  – If (Saturday or Sunday) and time is 8 am
Boolean Logic

- **OR** – at least one must be true or the expression is false.
  
  - True OR True = True; True OR False = True; False OR False = False
Boolean Logic

• AND – BOTH must be true or the expression is false.
  – True AND True = True; True AND False = False; False AND False = False

<table>
<thead>
<tr>
<th></th>
<th>True</th>
<th>False</th>
</tr>
</thead>
<tbody>
<tr>
<td>True</td>
<td>True</td>
<td>False</td>
</tr>
<tr>
<td>False</td>
<td>False</td>
<td>False</td>
</tr>
</tbody>
</table>
Boolean Logic

- NOT – negation. Flips True $\rightarrow$ False, or vice versa
  - NOT True = False; NOT False = True
How to Express the Algorithm

• Some people prefer a visual approach, such as a flow diagram:

Credit: jonkolko.com
Expressing the Same Algorithm in Words

• Others prefer to express in words:
Place phone call
if ringer is on
  repeat until user answers OR phone rings 4 times
    ring phone
  if user answers
    initiate conversation; when over end call
  otherwise if phone has rung 4 times
    leave message on voice mail and end call
otherwise leave message on voice mail and end call

Notice the importance of the indentation! You can include comments in your algorithm – typically using the # of // symbols to indicate the beginning of a comment.
Practice Algorithms

• Write an algorithm to do each of the following:

1. Input 3 numbers and print the average
2. Modify # 1 to print the average along with a second message indicating if the average was more than 100
3. Input 3 numbers and calculate the sum, but only include in the number in the sum if it is positive
Additional Practice

- Input the height and radius of a cylinder. Using these two values, calculate the surface area of the cylinder.
- Calculate a person’s age. Input the year born and if the user has already celebrated a birthday this year.