CMSC201
Computer Science I for Majors

Lecture 05 – Algorithmic Thinking

Last Class We Covered

- Decision structures
- One-way (using **`if`**)
- Two-way (using **`if`** and **`else`**)
- Multi-way (using **`if`**, **`elif`**, and **`else`**)
- Nested decision structures

Any Questions from Last Time?

Today’s Objectives

- To practice thinking algorithmically
- To understand and be able to implement proper program development
  - To learn more about “bugs”

- To get practice with decision structures
- (Lots of practice)

What is an Algorithm?

- Steps used to solve a problem

- Problem must be
  - Well defined
  - Fully understood by the programmer

- Steps must be
  - Ordered
  - Unambiguous
  - Complete

Developing an Algorithm

Program Development

1. Understand the problem
2. Represent your solution (your algorithm)
   - Pseudocode
   - Flowchart
3. Implement the algorithm in a program
4. Test and debug your program

Step 1: Understanding the Problem

- Input
  - What information or data are you given?
- Process
  - What must you do with the information/data?
  - **This is your algorithm!**
- Output
  - What are your deliverables?

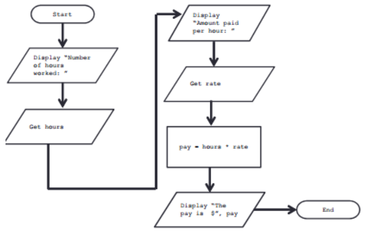
“Weekly Pay” Example

- Create a program to calculate the weekly pay of an hourly employee
  - What is the input, process, and output?

- Input: pay rate and number of hours
- Process: multiply pay rate by number of hours
- Output: weekly pay

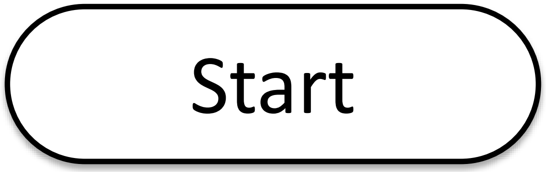
Step 2: Represent the Algorithm

- Can be done with flowchart or ***pseudocode***
- Flowchart

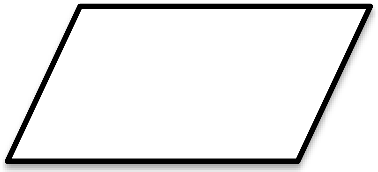

Start
Display "Number of hours worked: "
Get hours
Display "Amount paid per hour: "
Get rate
pay = hours * rate
Display "The pay is $", pay
End

  - Symbols convey different types of actions
- Pseudocode
  - A cross between code and plain English
- One may be easier for you – use that one

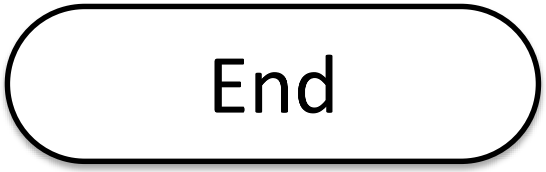
Flowchart Symbols



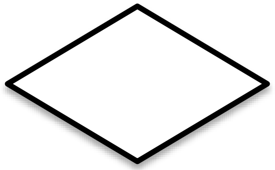
Start Symbol



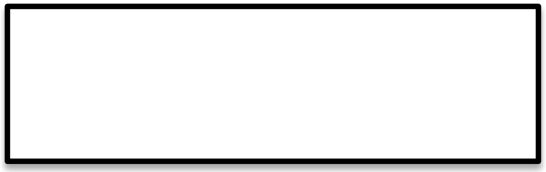
Input/Output



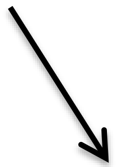
End Symbol



Decision Symbol

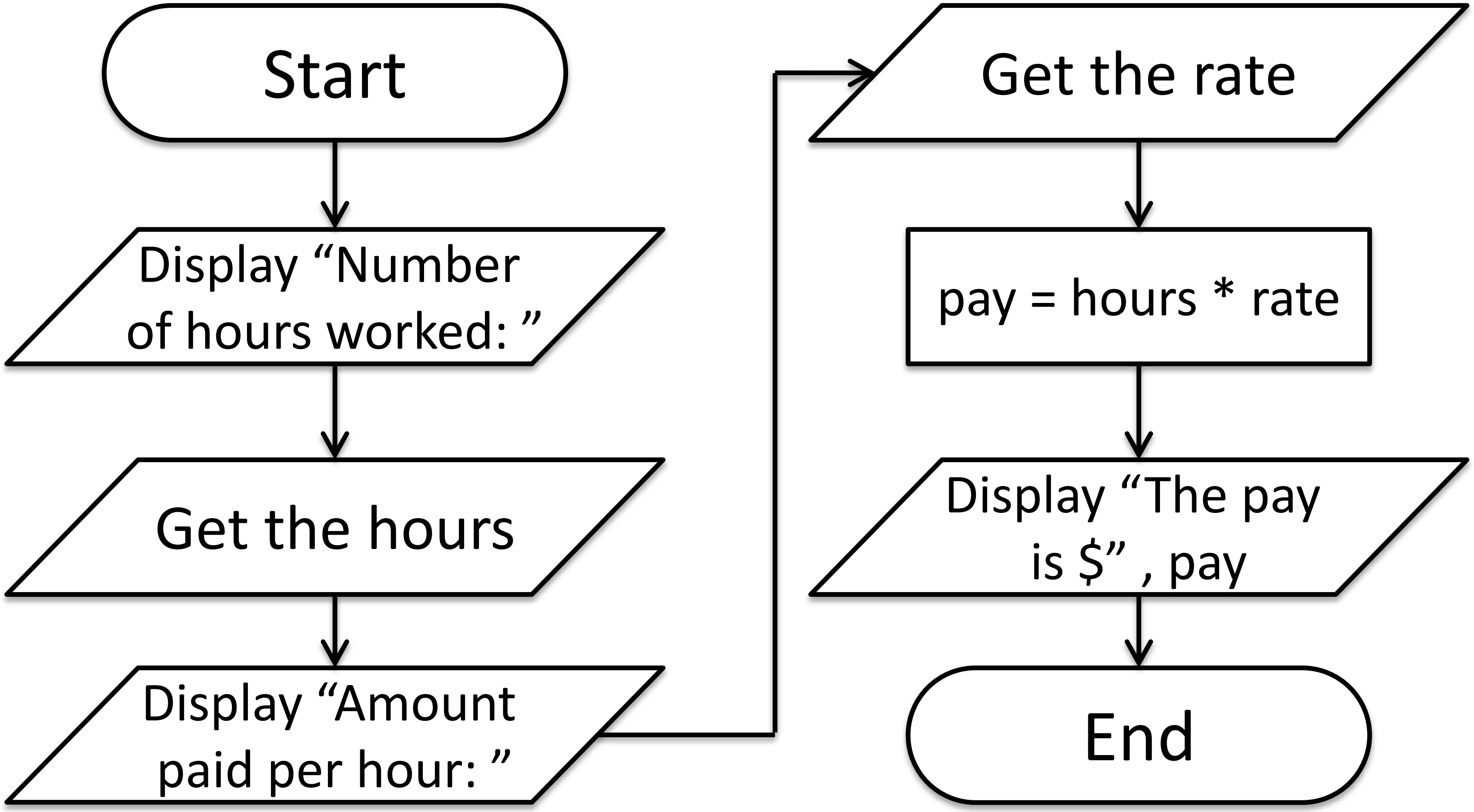


Data Processing Symbol



Flow Control Arrows

Step 2A: Flowchart


Start
Display "Number of hours worked: "
Get the hours
Display "Amount paid per hour: "
Get the rate
pay = hours * rate
Display "The pay is $" , pay
End

Step 2B: Pseudocode

- Start with a plain English description, then…

```
1. Display "Number of hours worked:  "
2. Get the hours
3. Display "Amount paid per hour:  "
4. Get the rate
5. Compute pay = hours * rate
6. Display "The pay is $" , pay
```

Steps 3 and 4: Implementation and Testing/Debugging

- Implementing and testing/debugging your program are two steps that go hand in hand

- After implementing, you must test it
- After discovering errors, you must find them
  - Once found, you must fix them
  - Once found and fixed, you must test again

Algorithms and Language

- Notice that developing the algorithm didn’t involve any Python at all
  - Only pseudocode or a flowchart was needed
  - An algorithm can be coded in any language
- All languages share certain tools that can be used in your algorithms
  - For example, ***control structures***

Algorithmic Thinking

- Algorithms are an ordered set of clear steps that fully describes a process
- Examples from real life?
  - Recipes
  - Driving directions
  - Instruction manual (IKEA)

Debugging

A Bit of History on "Bugs"

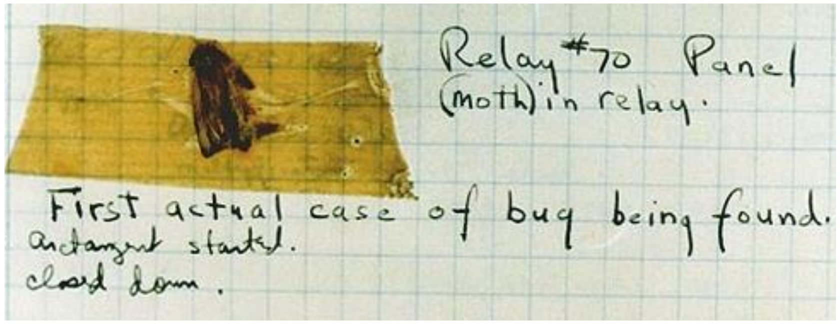
Rear Admiral Grace Hopper

- US Navy lab (Sep 1947)
- Grace Hopper and her colleagues were working on the Harvard Mark II
  - Instructions read one at a time from a tape
- Or trying to… it wasn't working right

A Bit of History on “Bugs”

Mark II, general view of calculator frontpiece, 1948.

- Mark II was a LARGE machine that took up an entire room
  - You could open each panel and look inside
- They found a literal bug inside the machine
  - Taped the bug (a moth) into their log book


Relay #70 Panel (moth) in relay.
First actual case of bug being found.
antangent started.
closed down.

Errors ("Bugs")

- Two main classifications of errors

- Syntax errors
  - Prevent Python from understanding what to do
- Logical errors
  - Cause the program to run incorrectly, or to not do what you want

Syntax Errors

- "Syntax" is the set of rules followed by a computer programming language
  - Similar to grammar and spelling in English
- Examples of Python's syntax rules:
  - Keywords must be spelled correctly

    `True` and **`False`**, not **`Ture`** or **`Flase`** or **`Truu`**
  - Quotes and parentheses must be closed:

    `("open and close")`

Syntax Error Examples

- Find the syntax errors in each line of code below:

```
prnit("Hello")
print("What"s up?")
print("Aloha!)
print("Good Monring")
```

Syntax Error Examples

- Find the syntax errors in each line of code below:

```
prnit("Hello")
print("What"s up?")
print("Aloha!)
print("Good Monring")
```

not actually a syntax error

Logical Errors

- Logical errors don't bother Python at all… they only bother you!
- Examples of logical errors:
  - Using the wrong value for something

```
currentYear = 2013
```

  - Doing steps in the wrong order
    - "Put the pan in the oven. Mix the wet and dry ingredients in the pan. Preheat the oven."

Practicing Decision Structures

Exercise: PB&J Algorithm

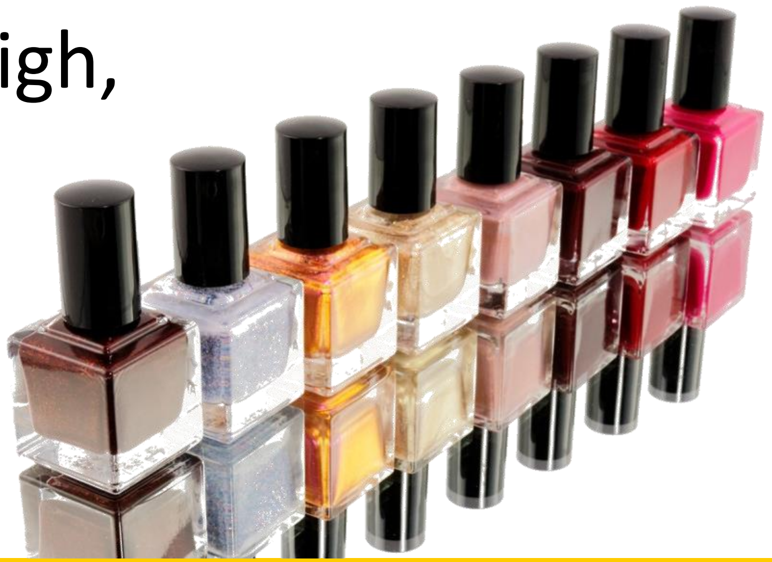
- English speaking aliens are visiting Earth for the first time. They want to know how to make a peanut butter and jelly sandwich.
- Explicitly, what are the required steps for building a peanut butter and jelly sandwich?

Exercise: Are Dogs Good?

- Ask the user if a dog is a good dog
- Print out one response for “yes”
- Print out a different response for any other answer

Exercise: Nail Polish

- Dr. Gibson has a LOT of nail polish
- Write a game where the user guesses how many bottles she has, and tell them whether their guess was high, low, or correct
- What info do you need?
  - (She has 296 bottles)



Exercise: Moving on to CMSC 202

- Ask the user their major and the grade they earned in CMSC 201
  - Print out whether they can move on to CMSC 202 next semester
- If they're a CMSC or CMPE major
  - They need an A or a B
- Otherwise
  - They need an A, B, or a C

Image from pixabay.com

Announcements

- HW 2 is out on Blackboard now
  - Complete the Academic Integrity Quiz to see it
  - Due by Friday (Feb 17th) at 8:59:59 PM
- Make sure to spell the dog breeds correctly!
  - Will make it much easier for your TA to grade
- Pre Lab 4 Quiz will come out Friday @ 10 AM
  - Must be <u>completed</u> by 9 AM Monday morning