Computer Science Principles Overview

A not so professional lesson from a professional programmer ….

- pro·gram
- ˈprōˌgram,-grəm/
- *verb*
- gerund or present participle: **programming**

- provide (a computer or other machine) with coded instructions for the automatic performance of a particular task.

- write computer programs.

- input (instructions for the automatic performance of a task) into a computer or other machine.

- verb: **programme**;   3rd person present: **programmes**

- cause (a person or animal) to behave in a predetermined way.

*What is programming?

Programming Truths

- **Programming is a way of thinking, not a rote skill.** Learning about "for" loops is not learning to program, any more than learning about pencils is learning to draw.
- **People understand what they can see.** If a programmer cannot see what a program is doing, she can't understand it.
- **Programs execute faster than you can imagine**! Don’t be afraid to do something simple a million times instead of something really complex one time.

Computer Truths

- The computer is NOT smart, in fact it is dumb.
- The computer is obedient EXACTLY to it's commands.
- The computer never does the wrong thing.
- The computer(or any program anywhere) is the ultimate example of "garbage in = garbage out."

Brad's Tips

- You MUST break each action into what is REALLY happening, down to a very small level. Really THINK HARD about what happens – it will let you write the best code.
- Smaller is better. Why? Smaller code is easier to understand, and it is less buggy.
- On a team, you DO NOT own your code. Let others use it, change it, criticize it, etc. Team code belongs to the team.
- Reuse code! If somebody else has done it already, its just a waste of time for you to do it again!
- It almost always takes double the time you think to accomplish a good program.
- "Measure twice, cut once" is better. This means plan out what you want to happen first, then program the device to do it. It WILL be faster than going "rapid fire" coding.

Building Blocks Simplified

- Programs need “materials”
- Programs will perform “actions”
- Programs have to make decisions
- Programs recognize when steps are repeated, and do them over and over.
- Many small programs added together are absolutely the best way to make a large program.

Building Blocks

- “Materials” are called Variables & Constants
- “Actions” are exactly that, code blocks that do something.
- “Decisions” are used to make the code take more than one action, depending on conditions. These are “if – then” or “switch” blocks.
- “Repeated actions” call for a “loop”. It is much better to use loops for repeats, it makes the code smaller and easier to understand.

Variables and Constants

- Variables hold data, that CAN CHANGE. You can use them in your program to hold the result of an action.
- Example: Variable called “sum”. We want to add 2 + 3. So, in code, [sum = 2 + 3] ← this is an action statement. At the end, sum = 5.
- Constants are variables that NEVER CHANGE. They are useful in a large program when a something is used over and over.
- Example: Constant “NUMBER OF SPINS” = 10. Our program might be like this:
- [turn right for NUMBER OF SPINS]
- [turn left for (NUMBER OF SPINS / 2)]

Actions

- Actions are simple. Make the program do something.
- [add 2+3] ← Math is an action
- [car/robot/thing go forward for 3 seconds] ← action
- [wait for 2 seconds] ← delays are an action

Decisions, If - then

- [if my car is at the edge of the cliff]
-    [then STOP!!!!]

- What happens if not at the edge of a cliff?

Decisions, Conditional

- [while I’m not at the edge of a cliff]
-    [drive forward]

- [stop]

Loops, not a loop

- Drive in a square:
- [drive forward for 2 seconds]
- [turn right 90 degrees]
- [drive forward for 2 seconds]
- [turn right 90 degrees]
- [drive forward for 2 seconds]
- [turn right 90 degrees]
- [drive forward for 2 seconds]
- [turn right 90 degrees]

Loops, Count

- Drive in a square
- [do 4 times]
-   [drive forward for 2 seconds]
-   [turn right 90 degrees]

- MUCH smaller, less risky because its very defined in actions, easier to understand.

Loops, Conditions

- Drive in a square
- [until the “car” is back to its starting point]
-   [drive forward for 2 seconds]
-   [turn right 90 degrees]

- MUCH smaller, a little more risk than previous example (Why?), easier to understand.

Ready to program?

- We’ll do two examples:
- First is making 2 cups of coffee
- Second is eating a McDonald’s Happy Meal.

Two cups of coffee?

- What are the variables (materials) needed?
- Are there any decisions to be made?
- What about loops?

Ethan’s Coffee

- Making Two Cups of Joe
- 1. Using a percolator, fill it with water until it reaches two cups.
- 2. Add one tablespoon of coffee grounds to the basket.
- 3. Reassemble the percolator.
- 4. Start the brewing process by plugging in the percolator.
- 5. The coffee is ready when the indicator light comes on.

Kevin’s Coffee

- 1. Gather needed items
  2. Grind beans
  3. Place filter in the brewing basket
  4. Add .75 ounces of coffee ground
  5. Fill reservoir with 12 fluid ounces of water
  6. Turn it on

  How did Moses make coffee?

  Hebrews it!!!

James Coffee

- How to Make Two Cups of Coffee
- Reach arm outward.
- Grab handle to pull out lever for the coffee grounds.
- Pull lever out.
- Release lever.
- Place hand on counter.
- Move hand until it touches filter.
- Grasp filter.
- Pick filter up.

James Coffee cntd

- Move arm up.
- Move filter over basket.
- Lower arm until filter is in basket.
- Release filter.
- Move arm over ground container.
- Move hand until it is on top of ground container.
- Grasp lid.
- Take other arm and move it over until it touches the container.
- Grasp container.
- Turn hand on lid counter clockwise.
- Release grasp.
- Turn hand clockwise.

James Coffee cntd

- Grasp lid and turn counter clockwise.
- Pull lid off.
- With left arm raise container.
- Move container over filter.
- Lower right hand onto counter.
- Release lid.
- Move right hand over container.
- Lower hand into container.
- Grasp scoop.
- Pull hand out of container.

James Coffee cntd

- Move scoop over filter.
- Dump coffee grounds into filter.
- Move scoop over container.
- Put scoop into container.
- Rotate hand to scoop up coffee grounds.
- Pull hand out of container.
- Move scoop over filter.
- Dump coffee grounds into filter.
- Move hand over container.

James Coffee cntd

- Drop scoop.
- Place container down on counter.
- Move right hand over lid.
- Lower hand until it touches lid.
- Grasp lid.
- Move right hand over container.
- Lower lid onto container.
- Turn hand clockwise.
- Release.

James Coffee cntd

- Turn hand counter clockwise.
- Grasp lid.
- Turn hand clockwise.
- Release both hands.
- Raise hand in front of lever to the coffee grounds basket.
- Move arm forward to close lever to the filter.
- Lower hand until in front of coffee pot.
- Extend hand until it touches coffee pot handle.
- Grasp handle.
- Pull arm back.
- Move pot until under water facet.

James Coffee final!!!

- With other hand, reach over to the facet handle. Grasp handle
- Pull facet handle out to turn water on.
- When water level hits the two-cup mark on the pot, push back on facet handle to turn water off.
- Release facet handle.
- Take left hand and place over coffee maker. Move arm down until it touches the back half of coffee maker.
- Grasp the flip top lid to water reservoir.
- Rotate hand counter clockwise.
- Move right arm with pot over the opening.
- Rotate arm counter clockwise to pour water into reservoir.
- Rotate right are clockwise.
- Move are in front of coffee maker.
- Move arm forward until pot is in original place.
- Release right arm.
- Rotate left hand clockwise until water reservoir lid shuts.
- Bring arm in front of coffee maker.
- Extend one finger.
- Move hand forward until start button is pressed.

Graham’s Coffee

- Pour 2 tablespoons of coffee into the coffee pot
- add 2 cups of water
- put pot in coffee maker and turn it on
- Wait 3 minutes and take pot out.
- Turn off coffee maker.
- Pour coffee into 2 cups.

Emily’s Coffee

- Put a coffee filter in the coffee machine
- Put 4 scoops of coffee grinds in the filter
- Turn on the coffee machine.
- After the coffee has been brewed take to cups
- Pour the coffee into the cups
- Take some creamer, sugar, and flavoring. Add it to your coffee.
- Add some whipped cream to the top for an extra touch

Brad’s “x” Coffee

- Variables: coffee, milk, sugar, water, cup = EMPTY, filter, pot, stirrer
- Constant: NUMBER OF CUPS = 2

```
Loop for NUMBER OF CUPS
  Water = 1 CUP.  //only use 1 cup
  pot = water  //water goes into pot
  Put coffee in filter.
  Heat water in pot
  until water is hot
      wait
  end

  Coffee = pour water through filter into pot //now have coffee!

  until cup is almost full
       pour coffee into cup
  end
  pour small amount of milk into cup
  pour small amount of sugar into cup
  stir (cup, stirrer)  //function call to stir function

end //NUMBER OF CUPS LOOP

DRINK!
```

Eat Happy Meal

- Variables: hamburger, fries, apples, drink, toy.
- Is there more?
- What’s wrong with this as the program?

- Eat Hamburger.
- Eat fries.
- Eat Apples.
- Drink.
- Play with toy.
- Mom trashes toy.

Eat Happy Meal

- Variables: box, toy, hamburger/wrapper, fries/sack, apples/wrapper, drink, straw, etc

- Steps not to forget:
- Take food out of box
- Remove wrappers
- Drink decision after each food
- One big loop?

Eat with Loop and decisions

```
•  One way:
•  Open box.
•  Remove all food from wrappers, put on plate.
•  Put straw in drink.
•  Until "all food is gone"
•    if hamburger is left
•      take small bite of hamburger
•    if a fry is left (ooh, why did I say it this way)
•      eat one fry
•    if an apple slice is left
•      eat apple slice
•    if thirsty AND drink is left (advanced concept here)
•      take a drink

•  Trash containers, play with toy
```

Code blocks

- Remember I mentioned sharing code, and smaller is better?
- Let's try code blocks, and see where that leads us

Eat Hamburger

- Remove wrapper from burger
- Until hamburger is gone
-   take a bite of hamburger

Eat Fries

- Remove fries from wrapper
- Until fries are gone
-   eat 1 fry ← why only one?

Eat Apples

- Remove apples from wrapper
- Until apples are gone
-   eat 1 apple slice

Drink

- Put straw in drink
- Until drink is gone
- take small sips of drink

Eat Happy Meal

- What's wrong with this as the program?
- Maybe everything – maybe nothing.

- Eat Hamburger.
- Eat fries.
- Eat Apples.
- Drink.
- Play with toy.
- Mom trashes toy.

Let’s have fun learning how to program!

- Primary goal is to understand coding fundamentals.
- There will be LOTS of repetition to reinforce learning
- We will learn as a team, work as a team, help each other as a team.
- Secondary goal is to understand many career aspects of programming in case you might want to pursue that one day