CS105 Problem Solving

Input / Output

Variables

If Statements

Wholeness

- In this lecture we will look at the basics of writing algorithms for the computer. Everything the computer does is because someone sat down and wrote an algorithm for it – told the computer how to do it.

- We will look at:
  - What are algorithms
  - What are variables
  - Input & Output
  - If Statements

Exercise: Writing an Algorithm

- An algorithm is a sequence of instructions to achieve a task
  - Cooking Recipes are algorithms
  - Driving instructions are algorithms

- Write some instructions for how someone, who hasn't been in this building before could find the bathroom (starting from where you are sitting)

Self Referral

- Did you notice how this process was very self referral?
  - Writing the instructions mostly involved: How would I do that?
- Did you notice that there are different possibilities?
  - Leave the door open, or close it when you leave?
  - Describe the distance in feet, steps, or meters?
- These are some of the most important points:
  - There is never one solution / no “the solution”
  - Therefore don’t try to get “the solution”, just give “how would I do that’

Sequence

- Why is the sequence important?

Flow Chart

- Flowcharts are a tool that allows people to express what the steps for a process are
  - http://xkcd.com/518/
  - https://xkcd.com/627/
  - http://xkcd.com/1195/

Make a Flowchart

- Lets write an algorithm to say “Hello World”
  - http://mumstudents.org/flowcharts/

- This flowchart tool is for expressing computer algorithms
  - Notice how there is an area for the variables and for the instructions
  - The instruction for changing a variable is called 'assignment' as a new value is assigned to the variable

Main Point 1

- An algorithm is a sequence of instructions. When people write an algorithm, they primarily think of 'how would I do that?'.
- Just like all of reality arises through the self interacting dynamics of the unified field, algorithms arise through the self interacting dynamics of our own consciousness

Data

- Would the algorithm you wrote be able to work on a computer?
    - Does a computer know what left or right is?
    - Does a computer know what or where a door is?

Nothing

- The computer doesn't know anything
  - About anything at all

- The computer can store things
  - Data items that its told to store
  - Algorithms (steps) aka programs

Model

- You have to create a model
  - Store data ‘about’ the world
  - An abstraction (just containing the things you need)

- Then have the computer manipulate it
  - Perhaps we have person on a map, and that person has X and Y coordinates
  - The person ‘moves’ by changing his X and Y

Model of the Building and of Our Person

- Lets make a drawing on this slide of a grid (x and y axis)
    - Draw the building layout, add my location, add the bathroom location

How do you Create a Model?

- What data do I need at the beginning?
- What data do I need at the end?
- What data will I manipulate to change the beginning to the end?
- (and what data will I need while manipulating?)

- It's always different, but always related to the task at hand. In our case:
  - X and Y of the person
  - X and Y of each segment of the walls

Variables

- X and Y are what we call variables
    - The computer manipulates (changes) them
    - To achieve the desired result

The Equals Sign

- Many programming languages use the equals sign for assignment (storage) into variables
  - This sometimes really confuses people up when starting CS
  - Doesn't make sense from a math background

- Important! The equal sign in most programming languages does not indicate equality
  - It indicates assignment

Exercise

What is the output of the following program

```
A = 10
B = 20
A = B
print A
print B
```

“Fix” the algorithm

- Lets think about the bathroom algorithm again
  - What would be our model
  - We would have to manipulate the X and Y of the person
  - Assuming a person cannot teleport
  - Assuming a person cannot walk through walls
  - At what X / Y would a person start?
  - At what X / Y would the bathroom be?
  - How big is each step?

Main Point 2

- The computer doesn't know about our world. In order for it to do anything we have to make a model, a fake world made out of numbers in the computer. We create many parts (variables) that it can change to achieve the desired result.
- The whole is greater than the sum of the parts

Useful

- Is our bathroom program useful so far?
  - X and Y of the person are only inside the computer
  - How does this tell a person where to go?

Input and Output

- In order for a program to be useful for humans, it needs:
  - Input: receive something (from a human)
  - Ouput: give something (to a human)

Hello World

- The traditional 'first program' for any language
    - Output “Hello World”
    - As shown earlier in the lecture

- Please note that output text needs to be in quotes
    - We actually output an expression
    - Expressions can contain other instructions as well!
    - How the computer knows what is what:
        - Instructions are words outside of quotes
        - Inside quotes is text (data)

Input and Output

- Next we'll update our program to take input
  - Our name

- And then output
  - Hello [name]

- We will do this by using:
  - The concatenation operator + to combine strings (text data)
  - The variable where the name is stored in our expression

Exercise: Expand the Program

- Create a flowchart that asks for a person’s name and age
    - First output “Hello “ + name
    - And then output “Your age is: “ + age
    - Notice both outputs go on the same line

New Lines

- What if we wanted to output 2 lines (1 for name, 1 for age)
- The computer has Special Control characters
  - These do not show up as text
  - Influence <u>how</u> the text is shown

- The most common of these is the newline
  - This is written as \n
  - The \ before it indicates that this is not a normal n
  - Because \ indicates the start of a control character, the only way to output an actual \ is by typing \\

Manipulate input
Create new output

- The purpose of basically every computer program is to take some input and turn it into the desired output

- The problem we as programmers solve is figuring out what the instructions and data model need to be.
  - In order to change the given input to the desired output

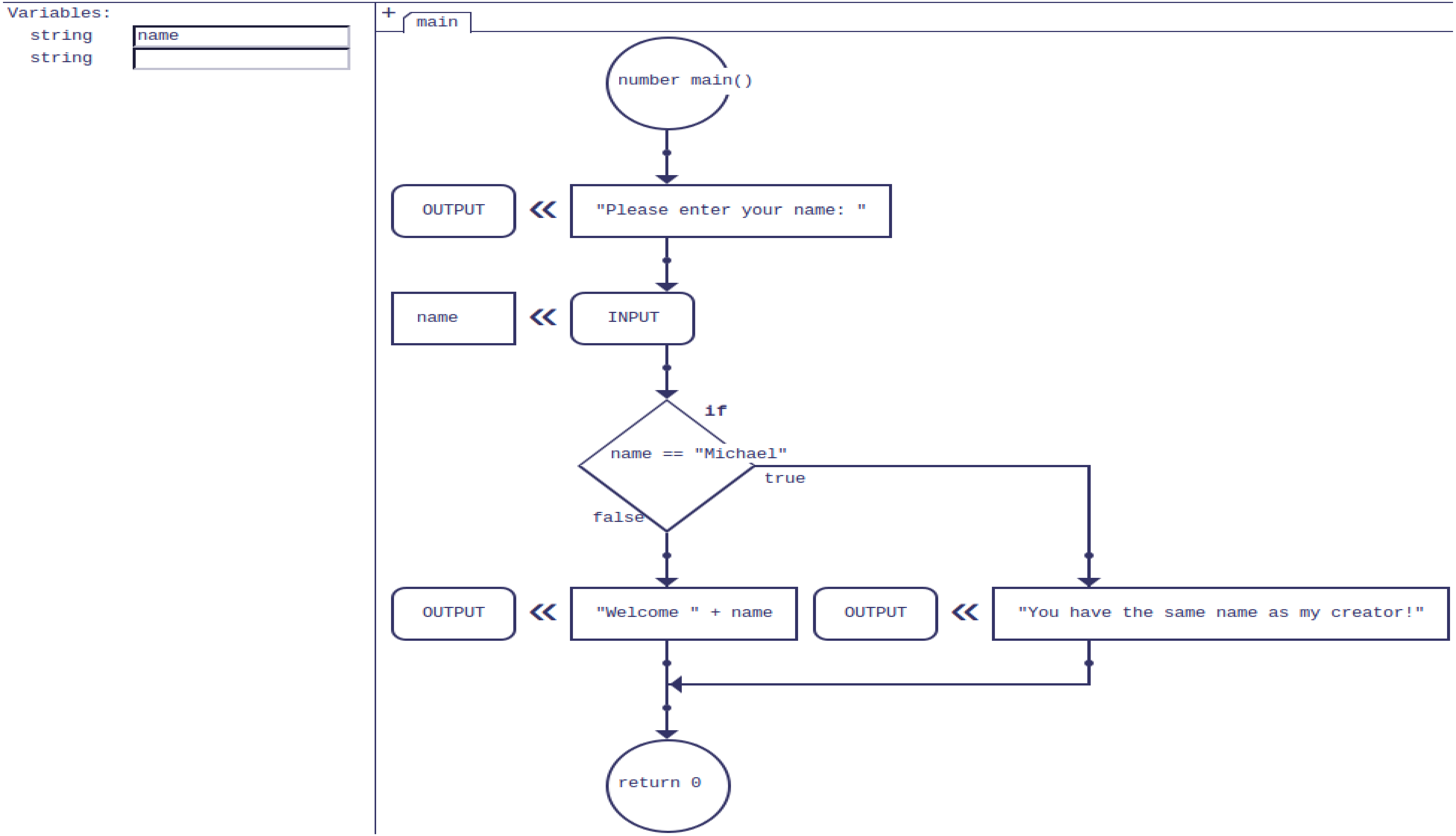
Main Point 3

- Input and Output are how the computer interacts with humans.

- In general every action has an equal and opposite reaction. Our goal is to create the optimal reaction based on the users action.

If Statements

- With input comes uncertainty
  - The user can input anything!
- Our program needs to respond correctly depending on what exactly the user entered
  - For now assume that the user enters correct data
- To respond differently to different inputs we use the If statement, which allows us to ‘branch’ the execution path



Variables:
string name
string

Decision Diamond

- The decision diamond has to contain an expression that will evaluate to true or false
- To create true or false you have to compare
- The following comparison operators exist
  - Equals: ==
  - Not equals: !=
  - Less than: <
  - Greater than: >

Exercise

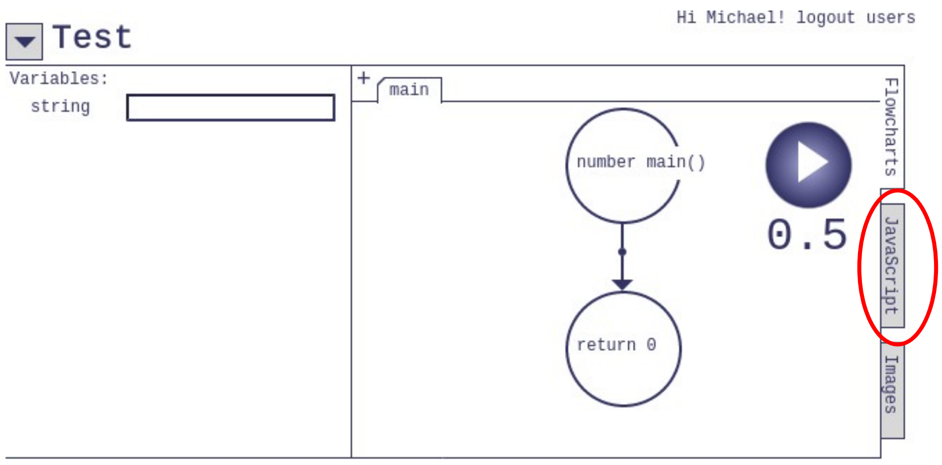
- Add an if statement to your flowchart that checks if the age > 16
    - If it is output “You can drive a car in Iowa”
    - Otherwise output “You are not allowed to drive”

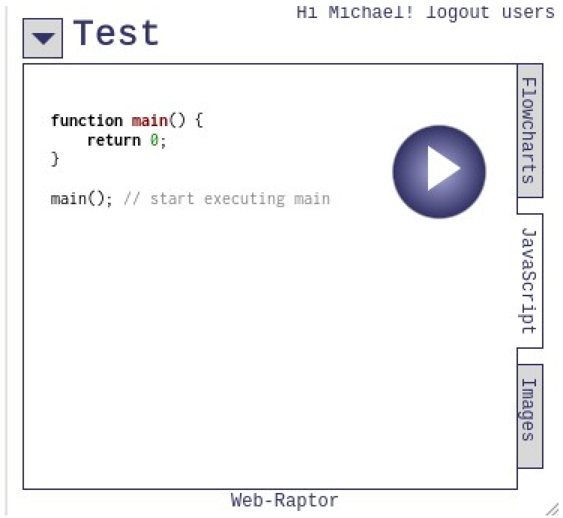
Main Point 4

- If statements allow us to branch a program so that different things happen under different circumstances
- Not just does every action have a reaction, we also have to ensure that right action is taken so as to create the correct results.

Reading and Writing Code

- Flowcharts are a tool for us to learn how to make an algorithm
  - The goal is to learn to read and write code
  - We will also practice reading and writing code on the daily quizzes
- Every flowchart can be turned into JavaScript code





```
function main() {
    return 0;
}

main(); // start executing main
```

Homework

- For your homework you’ll submit code
  - You’ll create a flowchart, but then submit the JS made from it
  - Be sure to read the JS before you submit!

Summary

- In this lecture we’ve seen:
  - What are algorithms
  - What are variables (model)
  - Input / Output
  - If Statements