README

- See sample README
  - You should have a README for EACH program
- Yours does NOT have to look exactly like this- it's just an example
- You can look up other examples online

Problem Solving

1325-Object-Oriented Programming

Fadiah Qudah

University of Texas at Arlington

Lecture Overview

- Lecture
  - Problem Solving (Foundations)
    1. Variables and Equations
    2. Reducing a Prompt/Problem
    3. Looking up Info
- Sample Programs

LECTURE

Problem Solving

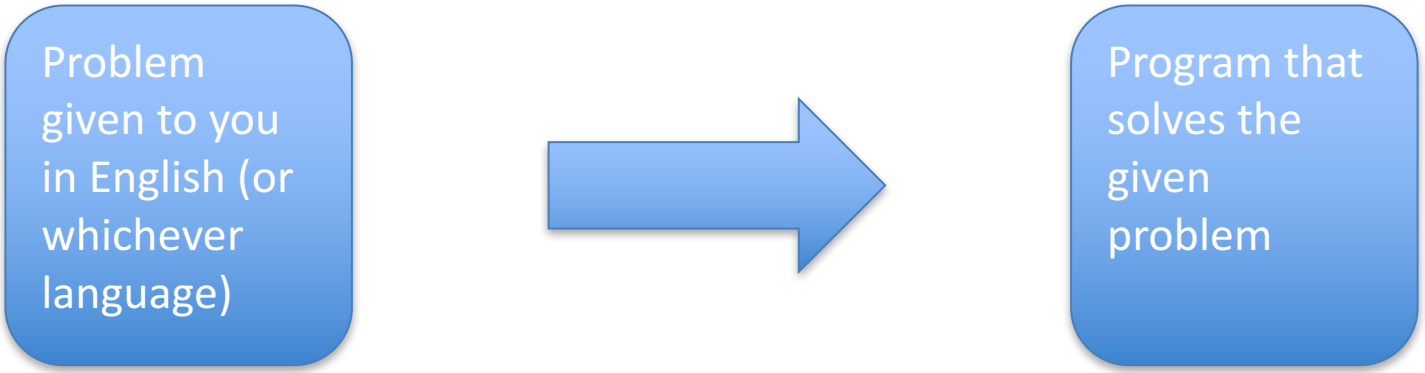
- The idea of problem solving is key to being a successful computer scientist
  - Clients wont tell you what they want
  - You need to explain the world around you to the computer
- It is the problem solving you do before hand that actually gets implemented into code
  - Code is pointless without some “thought” beforehand
  - It’s like someone rambling on and on to you incoherently

Problem Solving

- Today we are going to talk about translating a word problem into an equation
  - Most problems can be thought of as data/info and actions on that data/info
  - We'll be making equations using that data/info

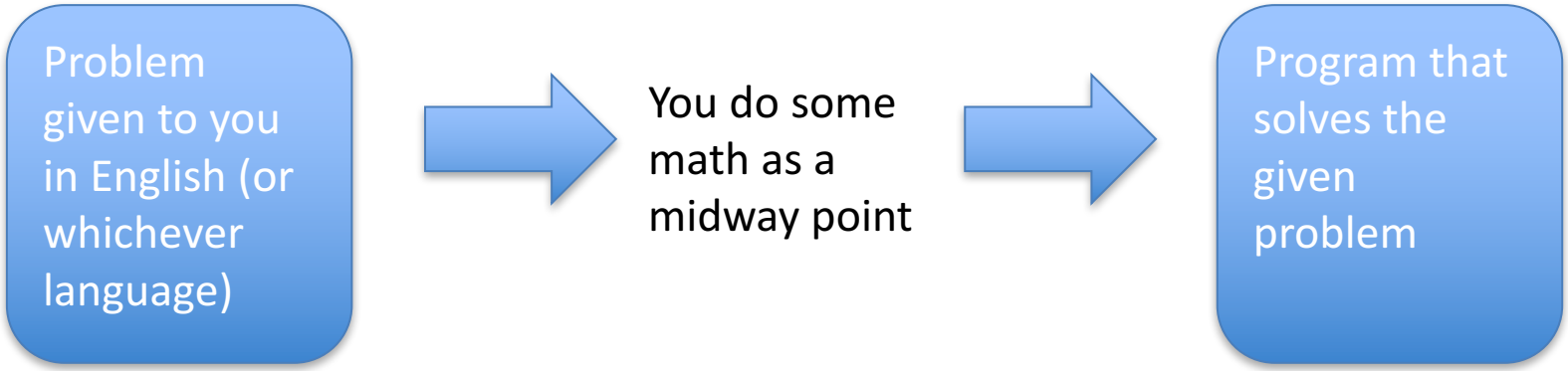
Problem Solving

- Turning English (or whichever language your problem is given in) to code
  - Example: A client telling you what they want a program to do



Problem Solving

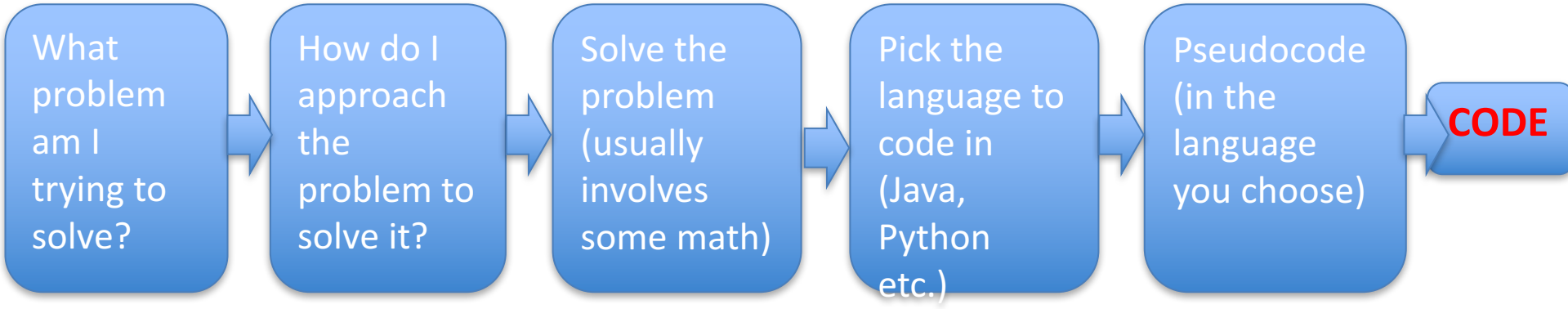
- Turning English (or whichever language your problem is given in) to code
  - Could also be a problem you observe yourself



Problem Solving

- Remember, the computer is a tool we use to solve some real world problem
- Our program is simply a list of instructions to the computer so it can solve our problem
- In order to tell our computer what to do, we ourselves need to know what we want to do

Problem Solving



- I like to follow the approach shown above when coding
  - Notice coding is the last step

1. **Variables and Equations**
2. Reducing a Prompt/Problem
3. Looking up Info

Variables and Equations

- What is an **expression**?
  - A math phrase containing values (numbers, variables) and operators
  - Evaluates out to some value:
    - 3x+4 4y-2s $7a+2^3$ b
  - Examples: The number of students in a class (3x), the weight of an elephant (2e+4), the age of a person (12)
  - In your programs, **you can hold the value of expressions in variables**
    - Info/data I talked about last time
    - int a=3b+4; /*3b+4 is an expression*/

Variables and Equations

- What is an **equation**?
  - A statement of equality between two expressions
    - $3^4-27x+2y = 7a+4^b$

  - I am saying that the two expressions above are equal to each other
  - Using **properties of equality**, we can manipulate an equation
    - Because we assume it is true our equation is equal

Variables and Equations

- Properties of equality are like the rules I use to make any "moves" on my equation
- Every "move" you make must follow a rule
- For example, you are **not** allowed to do something like this to your equation:

$$2x = 2(3x+5)$$

$$7000+yx+2x = 2(3x+5)^2$$

You can't just randomly add values to one side

You can't just randomly square one side

Variables and Equations

- Some commonly used properties (not all):

Addition Property: if a=b, then a+c=b+c 3x=2y -> 3x+7=2y+7

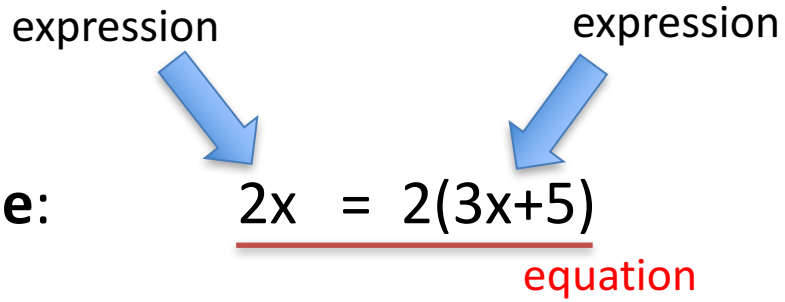
Subtraction Property: if a=b, then a-c=b-c 3x=2y -> 3x-5=2y-5

Distributive property: if $a(b + c) = ab + ac$ 3(x+2)=3x+6 *(6=3*2)*

Division Property: If $a = b$, then $\frac{a}{c} = \frac{b}{c}$ 3x=2y -> 3x/4=2y/4

(You can look up a full list)

Variables and Equations



Example:

I am saying these two expressions are equal. I can then manipulate this equation to find the unknown value of x.

Step 1:	2x=2*3x +2*5 2x=6x+10	Distributive property
Step 2:	2x-10=6x+10-10 2x-10=6x	Subtraction property
Step 3:	2x-10-2x=6x-2x -10=4x	Subtraction property
Step 4:	-10/4=4x/4 -10/4=x	Division Property

Variables and Equations

Let's now take a look at the following real world scenario:

Jane spent $42 for shoes. This was $14 less than twice what she spent for a blouse. How much was the blouse?

Variables and Equations

Jane spent $42 for shoes. This was $14 less than twice what she spent for a blouse. How much was the blouse?

We are looking for the value of our blouse. Since we do not know the value yet, let's make a variable, **b**, to represent the unknown price.

Using the variable b, let's convert this real world scenario into a mathematical equation (basically looking for quantities and operations on these quantities):

We will set our equation equal to 42 since the price of shoes equals everything else

less implies a subtraction operation

Jane spent **$42 for shoes**. This was **$14 less** than **twice what she spent for a blouse**. How much was the blouse?

Twice what she spent for b (our unknown blouse price) implies a multiplication operation on b

Variables and Equations

Jane spent $42 for shoes. This was $14 less than twice what she spent for a blouse. How much was the blouse?

42=2b-14

We can convert the given real world problem to the following math equation, using the variable b to represent the blouse price.

Variables and Equations

Jane spent $42 for shoes. This was $14 less than twice what she spent for a blouse. How much was the blouse?

42=2b-14

We can convert the given real world problem to the following math equation, using the variable b to represent the blouse price.

<u>We can use our properties of equality to solve for our blouse price, b:</u>

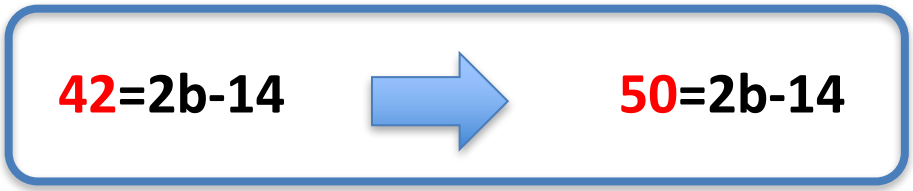
42=2b-14
42+14=2b-14+14 Addition property
56=2b
56/2=2b/2 Division property
28=b

Variables and Equations

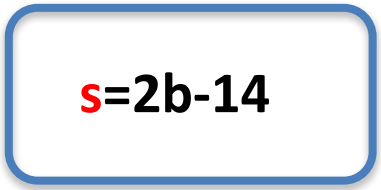
Now, what if we are also given the following scenario:

Gretel spent $50 for shoes. This was $14 less than twice what she spent for a blouse. How much was the blouse?

We can use the same equation as before, but now the total shoe price is 50 instead of 42:



How can we account for this? Instead of hardcoding the values 42 or 50, let's use a variable, **s**, to represent the shoe price:



By letting **s=42** OR **s=50** we can solve both scenarios given.

Variables and Equation

The same idea follows if want to handle this situation:

Sonya spent $60 for shoes. This was $20 less than twice what she spent for a blouse. How much was the blouse?

- We can handle the new shoe price of $60 since we have s as our shoe variable
- But now we're saying $20 less, not $14 less
- Lets create a new variable, called l:

Variables and Equations

Finally:

Zaza spent $70 for shoes. This was $10 less than three times what she spent for a blouse. How much was the blouse?

- Now we have to handle 3x as much, no longer 2x as much. Let's create a variable called t:

s=tb-l

We now have a general equation.

By generalizing the whole equation (and using variables), we can now have a computer program that can handle **any** of the preceding situations. We simply assign values to the variables accordingly.

Our program is not limited.

Variables and Equations

$s=tb-l$

Using the properties of equality, solve for b

Step 1: $s+l=tb-l+l$ Addition property

$s+l=tb$

Step 2: $s+l/t=tb/t$ Division property

$s+l/t=b$

This equation is what we will turn into code

Variables and Equations

Additionally:

We could say (we will talk about this next class):

– b is a function of s, l, t b=f(s, l, t)

(Remember, this just means that the value of b is dependent on the values of s, l and t)

Where f is the blouse price function

Note: I would only make this an actual function in my program if I do more than just use the equation

```
#include <iostream>

int main (int argc, char ** argv)
{

	int s=0,l=0,t=0,b=0;

	std::cout<<"Enter price of shoes: $";
	std::cin>>s;

	std::cout<<"Enter value less than blouse: ";
	std::cin>>l;

	std::cout<<"Enter value times blouse: ";
	std::cin>>t;

	b=(s+l)/t;
	std::cout<<"Blouse price is: $"<<b<<std::endl;

}
```

This code solves any of the previous blouse questions.

Notes:
-The more scenarios your code can handle, the better.

-Using variables instead of hardcoding

I just used this line to calculate the blouse price. I don't see a need to put this in a function. Someone else might decide to do so.

Variables and Equations

Generally for quantities:

triple, quadruple etc.

twice, four times, five times etc.

- If you see words like double or three times, you might be multiplying: Machine language instructions are quadruple the instructions you code.
  - int your_code_instructions=60;
  - int machine_instructions=your_code_instructions*4;

- The opposite of this is division: I made 3x what he made. (so made he made 1/3 of what I did)

  double my_salary=120000;

  double his_salary=my_salary/3;

Variables and Equations

- If you see words like gained, received, got, more than, greater than you might be adding:
  She received her first lotto check yesterday.

```
double money_in_bank, lotto;
money_in_bank=10.75;
lotto=500000.99;
money_in_bank=money_in_bank+lotto;
```

Variables and Equations

- If you see words like less than, lost, gave, spent, bought you might be subtracting: He lost some of his students after the drop date.

```
int student_total=12;
int student_dropped;
std::cout<<“How many students dropped? ”;
std::cin>>student_dropped;
student_total=student_total-student_dropped;
```

1. Variables and Equations
2. **Reducing a Prompt/Problem**
3. Looking up Info

Reducing a Prompt/Problem

- When given a prompt or real world problem (or sub-problem), it's a good idea to try and reduce the problem to the actual task you are trying to solve
  - It will help write a better program
  - It can also help with code reusability if multiple problems have the same task
    - We could write a function for the task and use it with different problems

Reducing a Prompt/Problem

- Jameson paints ceilings for $3 dollars/square foot. Create a program to help calculate the total price of a job.

- ABC Lawn will only mow lawns that 50 sq. ft or larger. Create a program to decide if a lawn can be mowed or not.

- Party Planners needs to know how many brochures are needed to cover a table. Create a program to help them with this.

(for simplicity: let the ceilings, lawns, tables and brochures be rectangular shaped for this example)

Reducing a Prompt/Problem

Even though the prompts seem completely different and unrelated, parts of what they are asking you to do are the same.

In all cases, you need to calculate the area of a rectangle at some point. You can make a function that does this for you and use it for all the problems.

- That way, we can reuse code (the function) instead of writing it again
- We can use the idea of variables (discussed in the previous section) and pass in different values that we need to calculate

Reducing a Prompt/Problem

Which ones are actually asking you to do the same task?

- See if at least one student made an A from a list of letter grades
- Given a list of yes or no answers (Y is yes, N is no) see if anyone said yes
- Given a list of letter grades, see if the total number of C students is less than 10

Which ones are actually asking you to do the same task?

- See if at least one student made an A from a list of letter grades

You are looking if a letter is present from a list of letters-yes it is or no it is not.

SAME TASK

- Given a list of yes or no answers (Y is yes, N is no) see if anyone said yes

You are looking if a letter is present from a list of letters-yes it is or no it is not.

SAME TASK

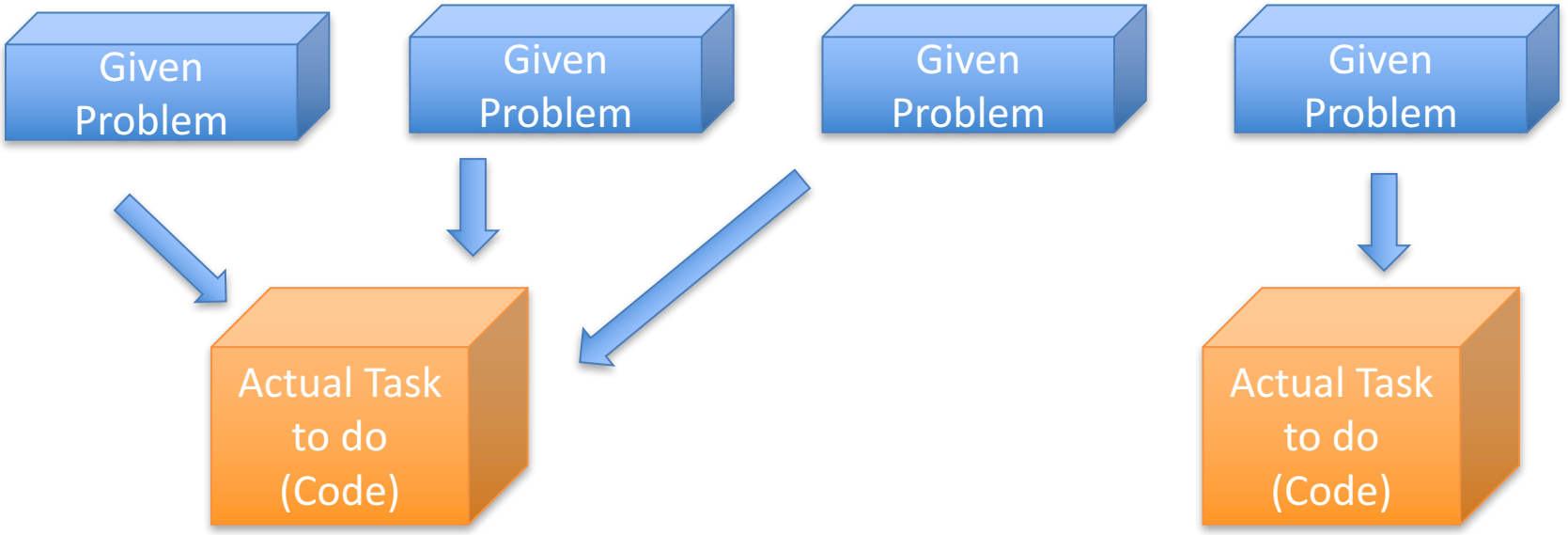
- Given a list of letter grades, see if the total number of C students is less than 10

DIFFERENT TASK

You are counting the number of times a letter appears in a list of letters-you will end up with an integer

Reducing a Prompt/Problem

- Even though tasks may look similar on the outside (same topic or same objects used), it is important to try to reduce to what you are **REALLY** trying to do



Reducing a Prompt/Problem

- Note that if you use different data structures to hold information, you may have no choice but to re-write code
  - If you use an array for one problem and a linked list for another, you will have to go through the data structure in a different manner.
    - This means the code won't be the same

1. Variables and Equations
2. Reducing a Prompt/Problem
3. **Looking up Info**

Looking Up Info

- Looking up information
- The concept of looking up is simple but important
  - I've noticed students get stuck when they are not directly given all the information in a problem
    - Worse, sometimes they blindly guess what something means
  - Being able to figure things out is the heart of computer science
- We can look up equations
- We can look up what defines something

Looking Up Info

- We can look up equations:
  - Farees drove from Arlington to Dallas, 25 miles away. He left Arlington at 7pm and arrived in Dallas at 7:30. He drove the same speed the whole time. How fast was he going?
    - We're looking for his speed.
    - We have an equation for that:
      - Speed=distance/time

Looking Up Info

- We can look up what defines something
  - Antoinette is a 13 year old girl living in Strasbourg. She wants to know how many more years until she can get her drivers license.
    - Look up-where is Strasbourg? France
    - Look up-what is the legal age to get a drivers license in France?
  - Texas might not have enough funding for the CHIP program. Would your family be affected?
    - Look up-What is CHIP?
    - Look up-What determines whether you will be affected? Family income level? Age?

Looking Up Info

- You may have some assignments where you will have to look up information
  - I will **NOT** specify to you that you need to look something up
  - I will of course **always** help with instructions and concepts for the assignments themselves
    - Help breaking up a problem, for example
  - Future clients will not always explicitly tell you what something is

SAMPLE PROGRAMS

Sample Programs

- See Class Code