Writing Pseudocode

Writing Pseudocode Video

0:01

When you're preparing to solve a problem using a program,

0:05

you should plan your program.

0:07

I like to tell my students that in the programming world,

0:12

as in the real world, it is ready, aim, fire.

0:17

Ready is understanding what you need to do,

0:19

which would be our problem defined here.

0:22

Aim is planning.

0:24

Fire is actually coding the program.

0:26

And you will find that things work out a lot better if you

0:29

do some planning.

0:31

So we've standardized on pseudocode,

0:33

which is a pretty common way to plan for a program.

0:37

And it helps if you think of pseudocode as a recipe.

0:43

Now there are no formal rules to pseudocode, whatever

0:46

works for you is pretty much OK as far as

0:48

indenting, capitalization, things like that.

0:51

It should serve the same function

0:53

as writing an outline before you write a paper,

0:56

helping to organize your thoughts

0:57

and know what you need to do.

0:59

I'd like you to use the recipe-type format.

1:02

So at the top, we'll have variables, which in your recipe

1:06

would be ingredients.

1:09

And then we will have directions or steps

1:13

that you need to program, which is like ingredients--

1:18

I'm sorry, which is like your directions in a recipe.

Variables (Ingredients)

Steps (directions)

1:25

So let's look at this first program,

1:28

and I'm going to give you the pseudocode on this

1:30

and show you how I solve it, so that you can then go forward

1:34

and actually program the program itself.

1:39

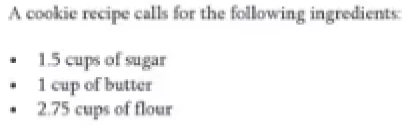
So we have a cookie recipe which calls for 1

1:42

and 1/2 cups of sugar, 1 cup of butter,

1:46

and 2 and 3/4 cups of flour.



A cookie recipe calls for the following ingredients:

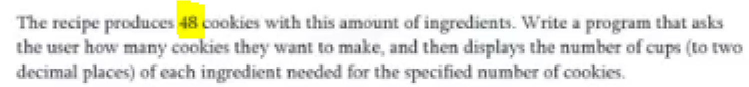
- 1.5 cups of sugar
- 1 cup of butter
- 2.75 cups of flour

1:49

And this recipe produces 48 cookies

1:52

with this amount of ingredients.



The recipe produces 48 cookies with this amount of ingredients. Write a program that asks the user how many cookies they want to make, and then displays the number of cups (to two decimal places) of each ingredient needed for the specified number of cookies.

1:54

So we know that we're going to need num cookies,

2:05

and we'll need to determine sugar, butter, and flour,

2:18

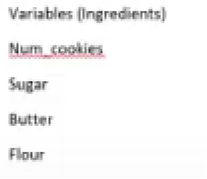
because that's what we need to determine.

2:20

So we know that those are our ingredients

2:24

or our variables in this case.



Variables (Ingredients)

Num_cookies

Sugar

Butter

Flour

2:27

Now ignore the fact that it capitalizes, that's fine.

2:29

It doesn't matter.

2:30

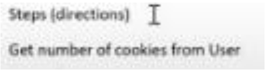
I wouldn't capitalize my variables.

2:32

So the first step for me would be get number of cookies

2:41

from user.



Steps (directions)

Get number of cookies from User

2:48

And then we need to figure out the same formula

2:55

based on sugar, butter, flour.

2:59

So we can do this a few different ways.

3:04

There's a lot of different approaches here.

3:05

So if you have 48 cookies for 1.5 cups of sugar,

3:14

we can figure out--

3:17

and we can actually set this up here.

3:19

This is one approach.

3:21

There is more than one-- there is more than one way

3:23

to do this.

3:25

1.5 divided by 48, that gives us our base amount.

3:39

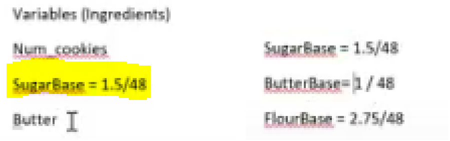
Butter would equal 1 divided by 48.

3:56

2.75 divided by 48, so that would give us

4:02

the actual ingredients for one cookie.



Variables (Ingredients)

Num_cookies

SugarBase = 1.5/48

Butter

SugarBase = 1.5/48

ButterBase= 1 / 48

FlourBase = 2.75/48

4:05

And so we could create the others

4:08

by-- we can get the sugar needed,

4:19

butter needed, flour needed, multiply.

SugarNeeded

ButterNeeded

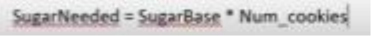
FlourNeeded

4:35

And we can just do this like code itself.

4:37

We can have sugar needed equals sugar base times num cookies,



SugarNeeded = SugarBase * Num_cookies

5:01

and I've sort of been doing this traditional Java-style

5:05

using Camelback notation.

5:07

And again, for pseudocode, it doesn't matter,

5:10

but the compiler likes us to use words separated by underscores.

5:25

So it can be better to get in that habit even

5:28

in your pseudocode.

5:30

I've been doing a lot of Java programming lately so--

5:33

so sugar needed equals sugar base,

5:42

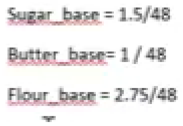
and I would actually be using lower case letters for all

5:45

of this, It's just that the Word likes to adjust that for me

5:49

automatically.



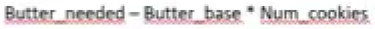
```
Sugar_base = 1.5/48

Butter_base= 1 / 48

Flour_base = 2.75/48
```

5:49

So we'll have butter needed equals butter

5:58

base times num cookies.



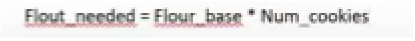
```
Butter_needed – Butter_base * Num_cookies
```

6:04

And we'll have flour needed equals flour

6:15

base times num cookies.



```
Flout_needed = Flour_base * Num_cookies
```

6:18

And again, there are other ways to solve this.

6:21

Most programming solutions, there's multiple approaches to.

6:24

So that would give us the numbers that we'd need.

6:28

Then we'd need to print output formatted

6:35

because we need to remember to format our numbers.

6:39

And we don't need to break those steps down.

6:41

If we know how to do it, we don't

6:43

need to break their steps down in the pseudocode.

6:48

This is just sort of high level logic.

6:50

So we've got our variables, and we've got our steps.

7:03

And you might find it neater if you indent these,

7:10

but these are all your own personal style and taste.

7:14

I don't have any formal way that I recommend doing pseudocode,

7:21

it needs to make sense to you.

7:23

But this would be the variables that I need and the steps.

7:26

And usually, I'm going back and forth between these two

7:29

because as I'm developing with steps,

7:32

I'll realize that I need an additional variable.

7:34

But this helps me when I go to program it

7:36

so that I can have all my variables, which typically

7:39

appear at the top of the code, and then I

7:41

can go through the steps.

7:43

And this is-- when we start handing in pseudocode,

7:45

this is the sort of thing I'm going

7:47

to look for you to hand in.

7:48

This is how you plan a program.

7:50

And so you can sort of use my plan here

7:52

to create and solve this first program.