

# pset 1: C

Zamyla Chan | [zamyla@cs50.net](mailto:zamyla@cs50.net)

# pset 1

- 0. A Section of Questions
- 1. Mario
- 2. Greedy

# Toolbox

---



- The Appliance
- Good style
- (Valid) user input
- Loops
  - ▣ For, While, Do-While
- printf
- Pseudocode

# The CS50 Appliance

---

- <https://manual.cs50.net/Appliance>
- gedit
  - ▣ text editor
- Terminal
  - ▣ run programs

# Navigating the Command Line

---

- ls
  - ▣ list the files in the current directory
- cd
  - ▣ change directories
- mkdir
  - ▣ make a directory
- rm [-r]
  - ▣ delete a file
- man
  - ▣ manual

# Style

---

- Indentation and {braces}
- Variable names
  - ▣ naming\_example
- Magic Numbers
- Comments

# Comments

```
// comments are really important  
// psst... Free points!
```

# // multi-line comments

---

```
/**
 *      Comment Template
 *
 *      By Zamyła
 *      CS50 2012
 *
 *      This gives a basic template for comments.
 *      See manual.cs50.net/Style Guide for tips!
 */
```



# Good Style vs. Bad Style

# Compiling and Running

---

- clang
  - ▣ 'translates' from C → machine code
- make

hello.c

# Mario



```
  ##  
   ##  
    ##  
     ##  
      ##  
       ##  
        ##  
         ##  
          ##  
           ##
```

# TODO

---

- Prompt for user input

apples.c

# TODO

---

- Prompt for ***correct*** user input

# While Loops

---

```
while (condition)
{
    // something happens
}
```

# Do-While Loops

---

- Executes the body *at least once*

```
do
{
    //something happens
}
while (condition);
```

# TODO

---

- ✓ Prompt for **correct** user input
- Draw the half-pyramid

```
##
###
####
#####
#####
#####
#####
#####
#####
```



# Find the pattern

- Height = 8
- First row: 2 hashes, 7 spaces
- Second row: 3 #, 6 spaces
- Third row: 4 #, 5 spaces
- ...
- $n^{\text{th}}$  row: how many hashes, how many spaces?
- Start at 0 or 1 indexing?

##

###

####

#####

#####

#####

#####

#####

# TODO

---

- ☑ Prompt for **correct** user input

- ☐ For every level of stairs...

  - ▣ print spaces

  - ▣ print hashes

  - ▣ print new line

##

###

####

#####

#####

#####

#####

#####

# For Loops

---

```
for (initialization; condition; update)  
{  
    // code inside  
}  
  
// code outside
```

# For Loops

---

```
for(int i = 0; i < height; i++)  
{  
    // print spaces  
    // print hashes  
    // print new line  
}  
  
// code outside
```

# Spot the difference

---

```
for(int i = 1; i <= height; i++)  
{  
    //...  
}
```

□ What about a while loop?

hello.c

# Choose carefully!

---

Conditions → Initializations ↓	$i < 10$	$i \leq 10$
$i = 0$	<b>Repeats 10 times</b>	Repeats 11 times
$i = 1$	Repeats 9 times	<b>Repeats 10 times</b>

Greedy

# Example

---

```
jharvard@appliance (~/.pset1): ./greedy
0 hai! How much change is owed? .32
4
```



# Thought Process

---

Input: \$0.32

1. Can I use a quarter? Yes.
2. Can I use another quarter? No.
3. Can I use a dime? No.
4. Can I use a nickel? Yes.
5. Can I use a penny? Yes.
6. Can I use another penny? Yes.
7. Can I use another penny? No.

4 coins used

# TODO:

---

- Prompt user for a monetary amount
  - ▣ The amount must make sense
  - ▣ What values are accepted?

# \$ to ¢

---

- Input is a value in dollars
- How to convert to cents?
- Floating point imprecision
- `imprecision.c`
- `round()`

# TODO:

---

- ☑ Prompt user for a monetary amount
- ☐ At each 'step,' return the biggest coin possible
- ☐ Keep track of:
  - ▣ how many coins to be returned
  - ▣ amount to be returned
- ☐ Print the final amount of coins.

# printf

---

```
printf("hello\n");
```

```
int count = 4;  
count = count * 2;  
printf("%d", count);
```

```
long long big_num = 2147483648;  
printf("I have %lld dragons", big_num);
```

# modulo math

□ % returns the remainder of the division

□  $10 \% 2 = 0$

□  $6 \% 5 = 1$

□  $7 \% 9 = 7$

□  $74 \% 3 = 2$

$$\begin{array}{r} 24 \\ 3 \overline{) 74} \\ \underline{6} \\ 14 \\ \underline{12} \\ 2 \end{array}$$

# Pseudocode

---

```
get amount in dollars
while (quarters can be used)
    increase count
    amount decreases by a quarter
while (dimes can be used)
    increase count
    amount decreases by a dime
(etc...)
print number of coins used
```

this was walkthrough 1