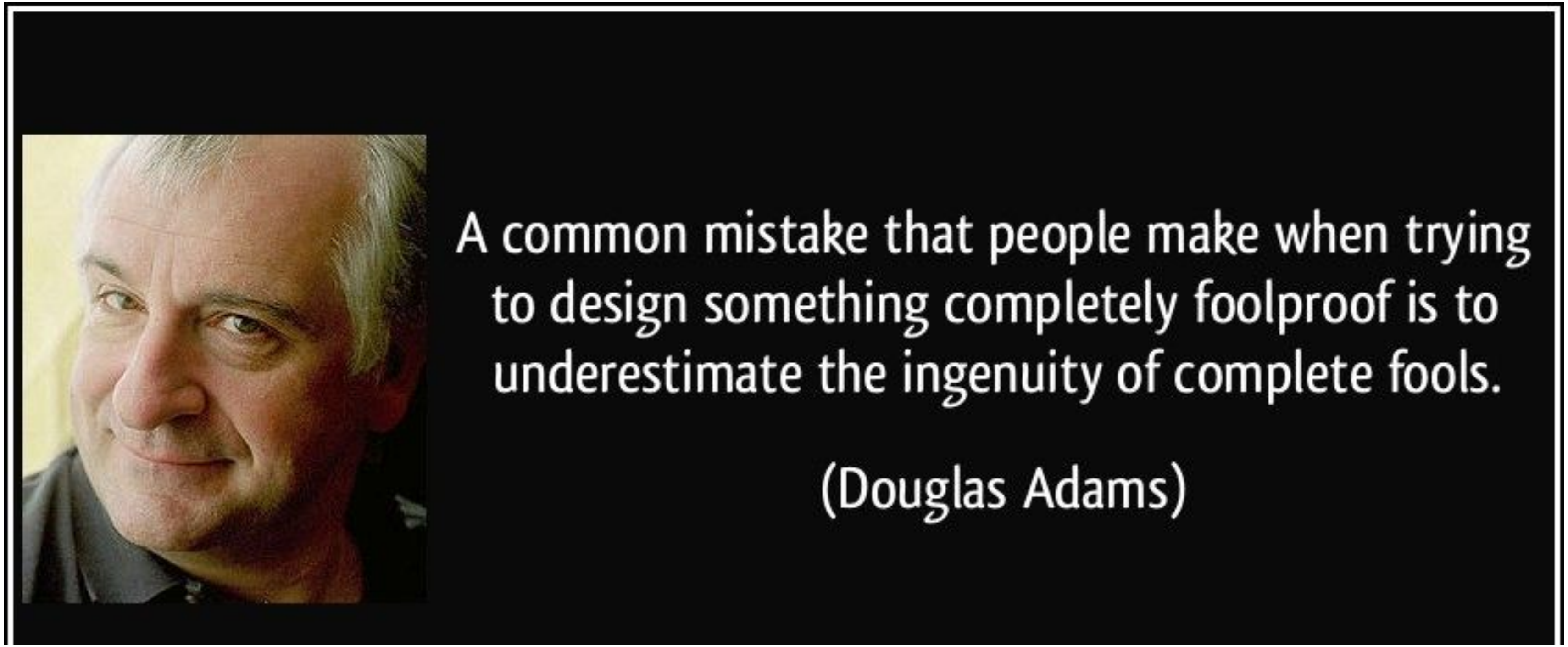
Input Verification

User Intelligence Level



A common mistake that people make when trying to design something completely foolproof is to underestimate the ingenuity of complete fools.

(Douglas Adams)

Input Verification

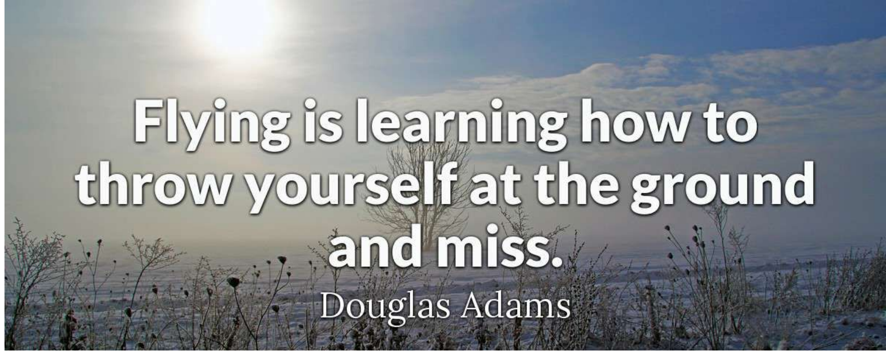
In the beginning the universe was created.

This has made a lot of people very angry, and has been widely regarded as a bad move.

- Douglas Adams -

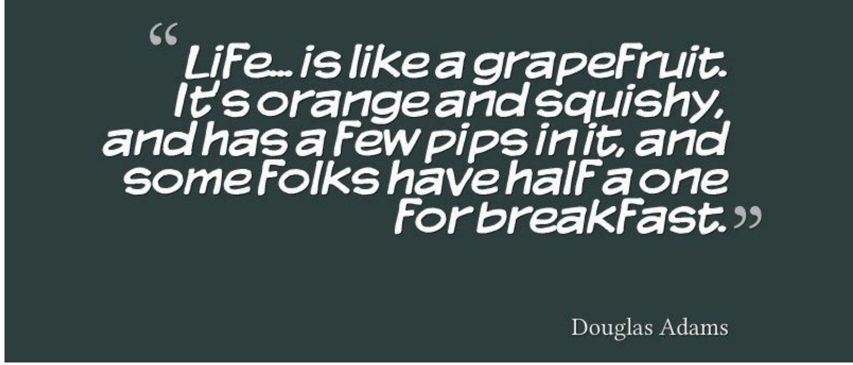
- Goal of software - make something as efficient as possible
  - Goal = efficiency of use as well
  - Question: How do people (who have no idea how to use your program) use your program efficiently and correctly?

Input Verification


Flying is learning how to throw yourself at the ground and miss.
Douglas Adams

- Question: How do people (who have no idea how to use your program) use your program correctly?
    - ANSWER: THEY’RE NOT GOING TO
    - A programmer must ensure mistakes do not break the code’s functionality

Input Verification

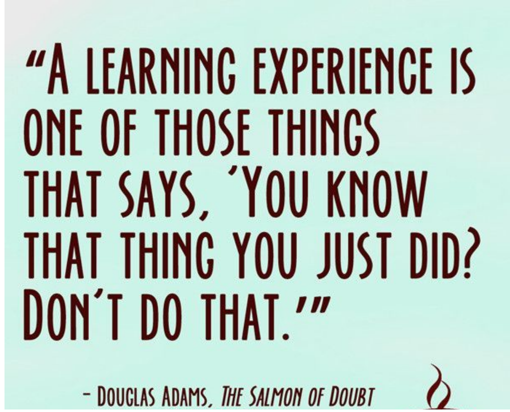


“Life... is like a grapefruit. It's orange and squishy, and has a few pips in it, and some folks have half a one for breakfast.”

Douglas Adams

- Input Verification
    - Designing ways to make sure users do not “break code”
    - Three main ways
        - Walk users through every step (*trusting*)
        - Do nothing with bad input (*middle ground*)
        - Ensure input is correct (*untrusting*)

Input Verification 1

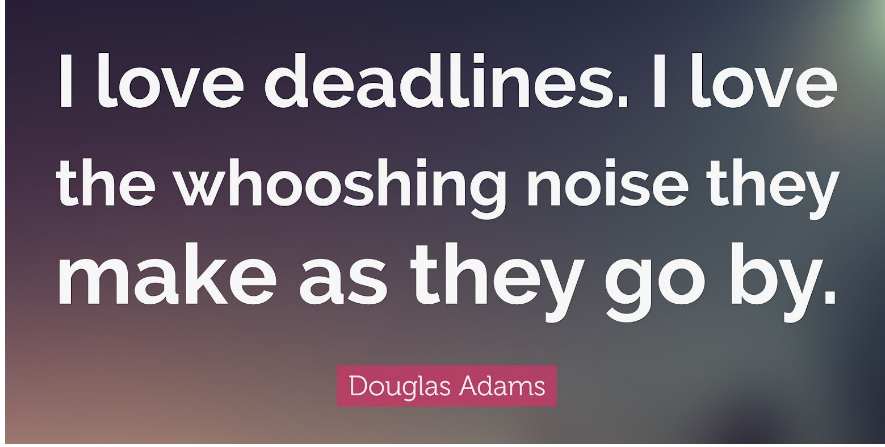


“A LEARNING EXPERIENCE IS ONE OF THOSE THINGS THAT SAYS, ‘YOU KNOW THAT THING YOU JUST DID? DON’T DO THAT.’”

– DOUGLAS ADAMS, *THE SALMON OF DOUBT*

- Walking users through a process
  - Tell users each step along the way
  - *Snake Code example:*
    - *Tell users at EVERY STEP to enter valid input*
      - *“Enter a ‘u’, ‘d’, ‘r’, or ‘l’:*

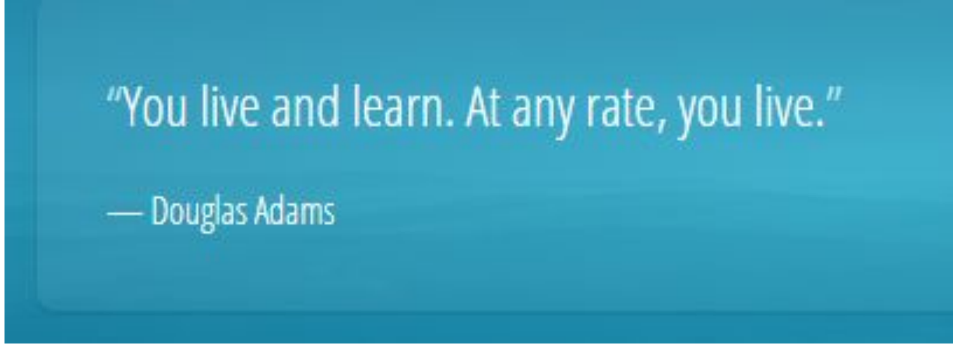
Input Verification 1


I love deadlines. I love the whooshing noise they make as they go by.

Douglas Adams

- Problems:
  - Users do not (or cannot) read instructions
  - Users misunderstand directions
  - Most important: Users ignore directions

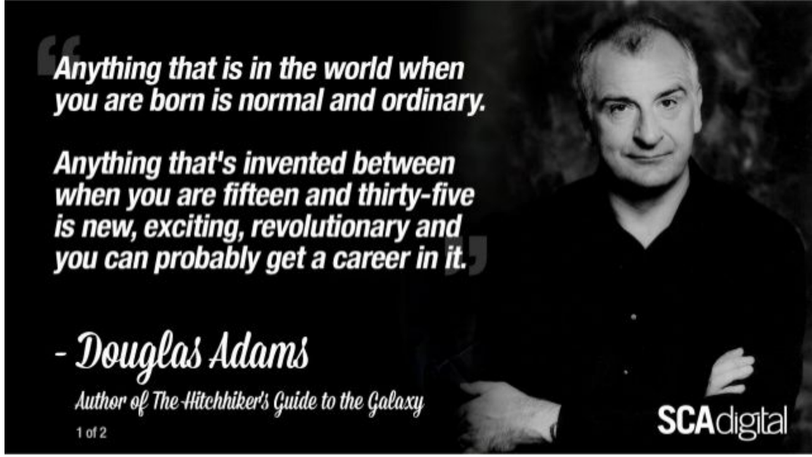
Input Verification 2


"You live and learn. At any rate, you live."

— Douglas Adams

- Do nothing with bad input
  - Your code only responds IF the user enters correct input (otherwise, it will remain the same)
  - *Snake example:*
    - *If the user enters something besides 'u', 'd', 'l', or 'r' → print the board again with no movement*

Input Verification 2



"Anything that is in the world when you are born is normal and ordinary.

Anything that's invented between when you are fifteen and thirty-five is new, exciting, revolutionary and you can probably get a career in it."

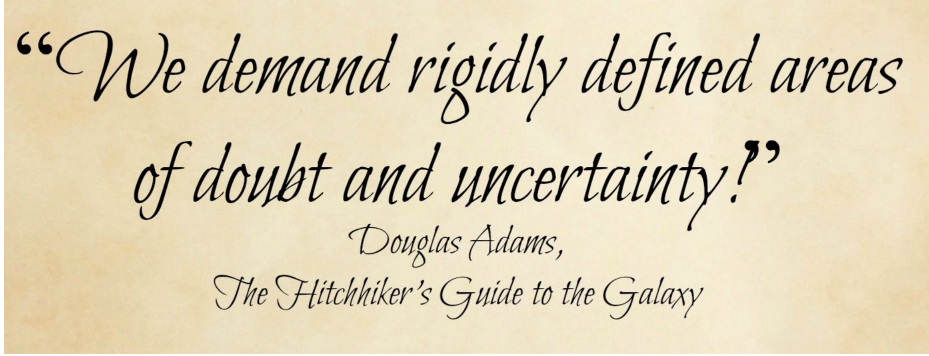
- Douglas Adams

Author of The Hitchhiker's Guide to the Galaxy

1 of 2

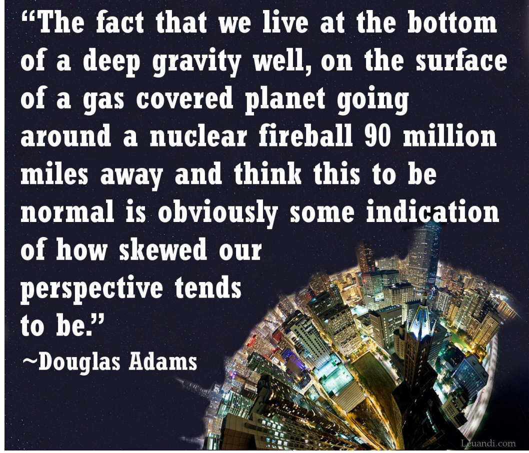
- Problems
    - If is not paired with #1, users are extremely confused
    - Can lead to problems in certain sequential situations

Input Verification 3


"We demand rigidly defined areas of doubt and uncertainty!"
Douglas Adams,
The Hitchhiker's Guide to the Galaxy

- Ensure that the user enters correct input
    - Gives warning signs if they did not, and continuously repeats until correct input is received
    - *Snake example:*
        - *WHILE the user does not enter 'u', 'd', 'l', or 'r', continuously prompt the user to enter 'u', 'd', 'l', or 'r', and do not move on until they do*

Input Verification 3


"The fact that we live at the bottom of a deep gravity well, on the surface of a gas covered planet going around a nuclear fireball 90 million miles away and think this to be normal is obviously some indication of how skewed our perspective tends to be."
~Douglas Adams
Leuandi.com

- Problems:
  - If instructions are unclear, the user can get frustrated

  - **Overall, this is the most common solution to user input**

Input Verification Coding Challenge

- Board challenge: Input two dimensions for a board, but those dimensions must be between 0 and 20
    - Use Strategy #3 to ensure that your code will function

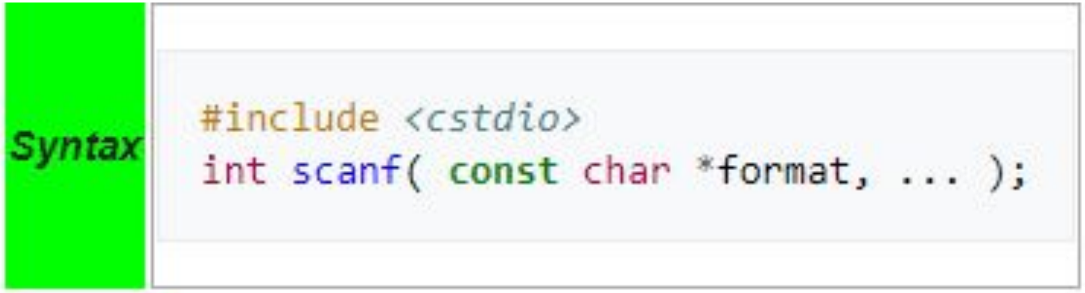
Input Verification Coding Challenge

- Board challenge: Input two dimensions for a board, but those dimensions must be between 0 and 20
  - Use Strategy #3 to ensure that your code will function

  - **WHAT ELSE CAN BREAK THIS CODE?**

Input Verification: Different Variable Types

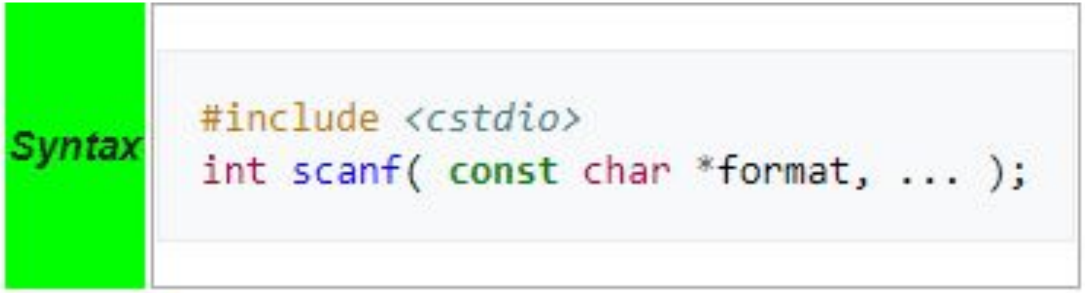
- What if the code is expecting integers, but the user enters a string?
- Scanf() is helpful here!



```
#include <cstdio>
int scanf( const char *format, ... );
```

Input Verification: Different Variable Types

- Scanf() returns 1 if input was scanned into the variable with no errors
    - Returns 0 otherwise



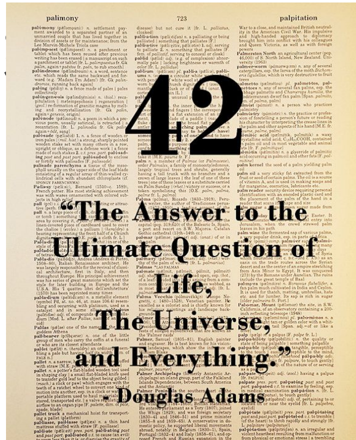
Syntax

```
#include <cstdio>
int scanf( const char *format, ... );
```

Input Verification: Different Variable Types

- To ensure that the user enters integer input...

```
while (scanf(“%d”, &num) != 1){
   printf(“Wrong input. Enter a number: \n”)
}
```


42
“The Answer to the Ultimate Question of Life, The Universe, and Everything.”
- Douglas Adams