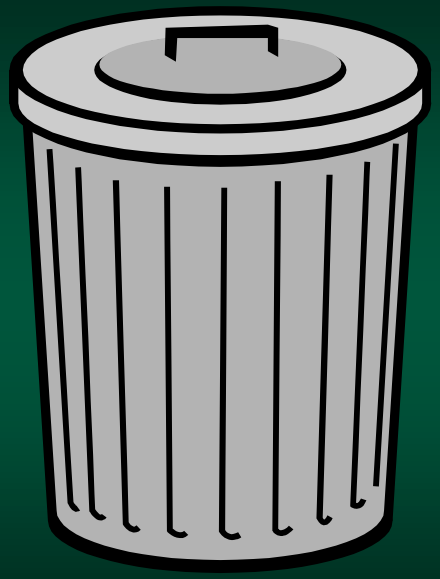
Input Validation

Chapter 7

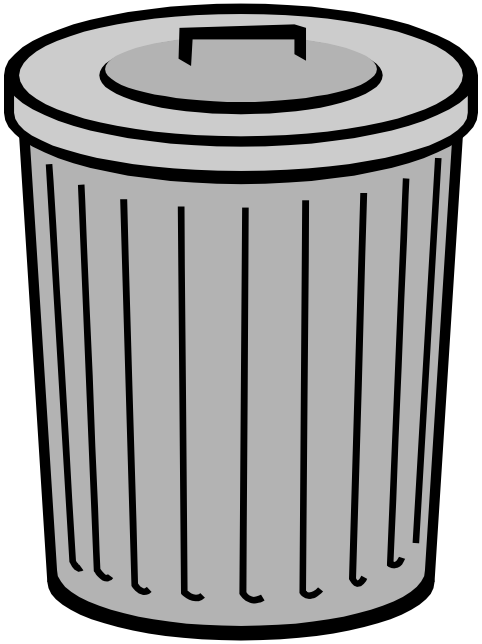


Garbage In, Garbage Out

Chapter 7.1

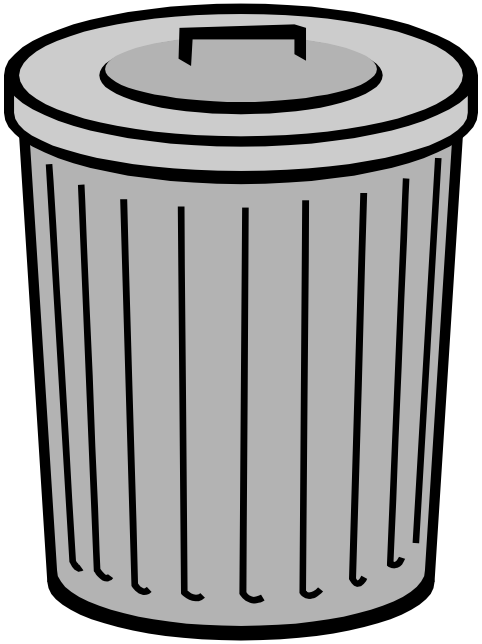
Garbage In, Garbage Out

- In computer science, the term *"garbage-in garbage-out"* refers to how bad input can create corrupted results
- If a program reads *bad* data as input, it will produce *bad* data as output



Garbage In, Garbage Out

- Often computer programs will abbreviate "garbage-in garbage-out" as *GIGO*
- Yes, it is so common it got an acronym!



Example: Pizza Party

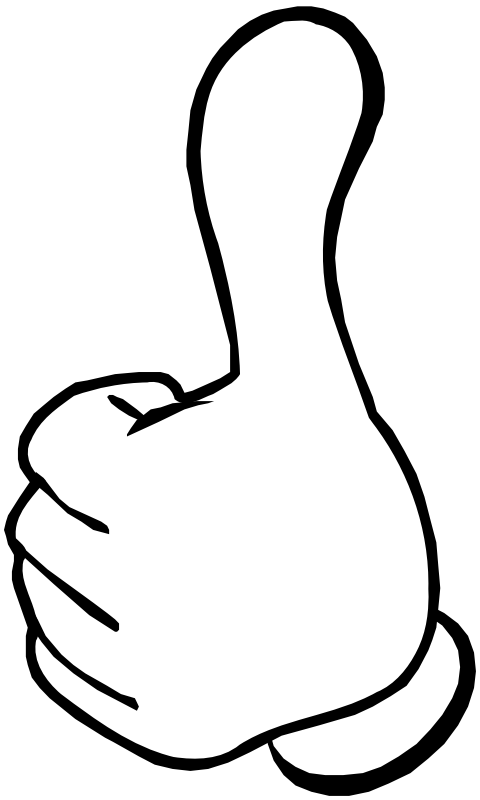
```
Declare Integer totalSlices, guests
Declare Real slicesEach

Set totalSlices = 64
Input guests

Set slicesEach = totalSlices / guests
```

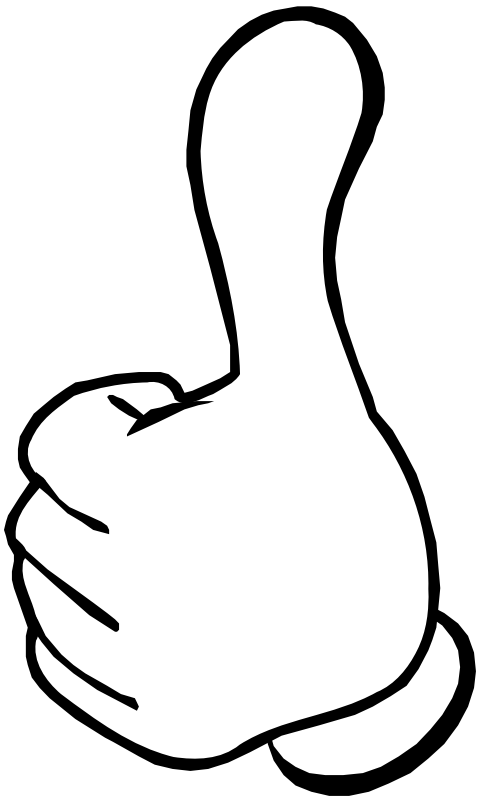
We can divide by zero or a negative value!

Garbage In, Garbage Out



- Programs should be designed to accept only good data
- Generally, programmers assume the user is going to make mistakes, and prepare for them

Input Validation



- *Input Validation* is used to prevent bad data
- All input should be inspected before processing
- If it's invalid, it should be rejected and the user should be prompted to enter the correct data

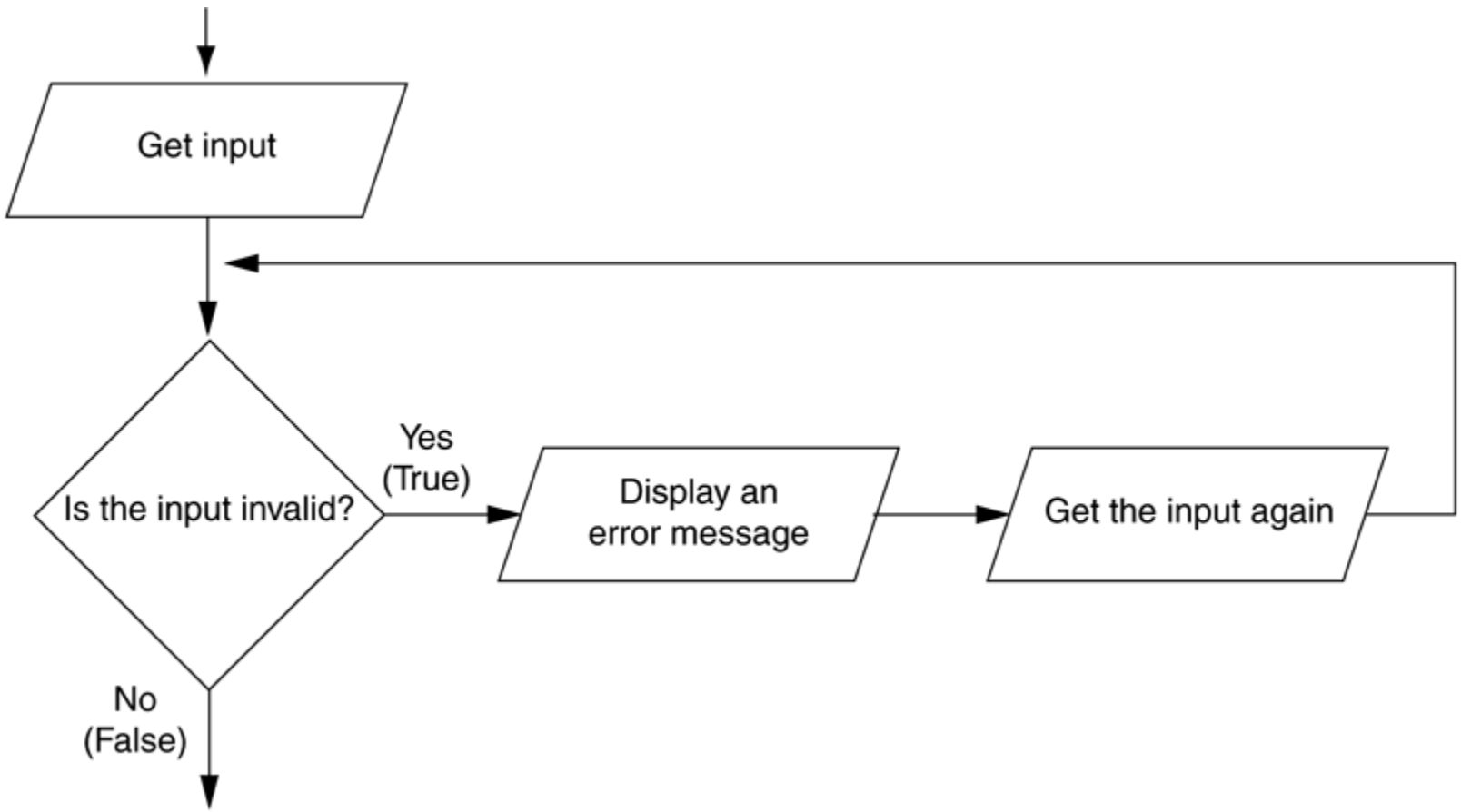
The Input Validation Loop

Chapter 7.2

The Input Validation Loop

- Input validation is commonly done with a loop that iterates as long as input is bad
- So, the user is basically "trapped" in the loop until they enter valid data
- The program will not proceed unless the data is good

The Input Validation Loop



The Input Validation Loop

```
Input score

While score < 0 OR score > 100
    Display "Invalid score!"

    Input score
End While
```

Priming read

Reenter value

Example: Pizza Party Guests

```
...

Input guests

While guests < 1
   Display "Enter a valid number"
   Input guests
End While

...
```

Trap user. Won't proceed until valid

Defensive Programming

Chapter 7.3

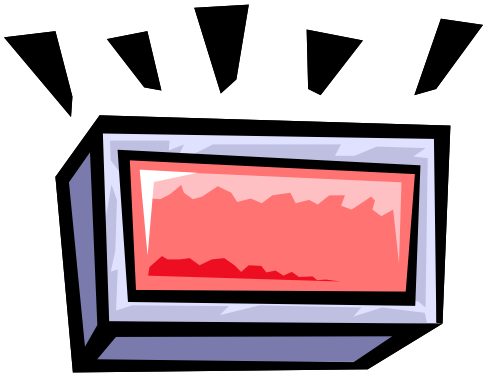
Defensive Programming

- *Defensive programming* is the practice of anticipating both obvious and <u>unobvious</u> errors
- Input validation is defensive programming

Types of Errors to Consider

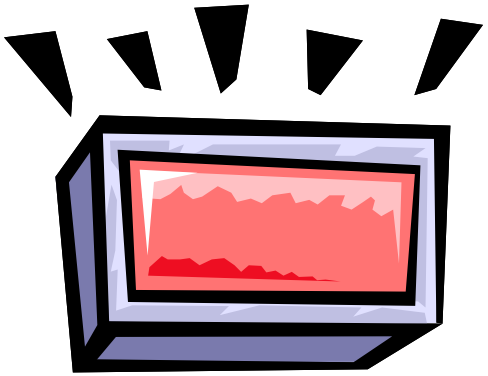
- Empty input, where a user accidentally hits enter before entering data
- The user enters the wrong type of data
- Values outside a valid range
- Data that is too long/short

Common Errors



- State abbreviations should be 2-character strings
- Zip codes should be in the proper format of 5 or 9 digits
- Hourly wages and salary amounts should be numeric values and within ranges

Common Errors



- Dates should be checked
- Time measurements should be checked
- Check for reasonable numbers