# CMSC201
# Computer Science I for Majors

# Lecture 06 – While Loops

# Last Class We Covered

- Algorithms

- Program design
  - Input, process, output
  - Flowcharts and pseudocode

- Decision structure practice

# Any Questions from Last Time?

# Today's Objectives

- To learn about and use a `while` loop
  - To understand the syntax of a `while` loop
  - To use a `while` loop for interactive loops


- To practice conditionals

# Announcement: BRAID Survey

- Based on your enrollment in this course, you will be receiving an email inviting you to take a short survey about your experiences in computer science. I strongly encourage you to participate in this survey, since your participation is vital to understanding the diverse experiences of computer science students.

- The first 400 respondents to complete the survey across the BRAID campuses will receive a $15 Amazon gift card as compensation and all students who complete the survey will be entered in a drawing to win one of two $125 Amazon gift cards.

- Please watch for an email with a link to complete the survey.

BRAID | Building Recruiting And Inclusion for Diversity

# Practice: **if**, **elif**, and **else**

```
if
else
elif
```

```
if
elif
elif
else
else
```

```
if
elif
if
elif
```

```
if
if
elif
else
```

```
if
    if
    if
else
    if
```

```
if
else
    if
    elif
    elif
else
```

```
if
elif
```

# Practice: `if`, `elif`, and `else`

```
if      ✗
else
elif
```

```
if      ✗
elif
elif
else
else
```

```
if      ✓
elif
if
elif
```

```
if      ✓
if
elif
else
```

```
if          ✓
    if
    if
else
    if
```

```
if      ✗
else
    if
    elif
    elif
else
```

```
if      ✓
elif
```

# Practice: `if`, `elif`, and `else`

```
if      ✘
else
elif
```

```
if      ✘
elif
elif
else
else
```

```
if      ✔
elif
if
elif
```

```
if      ✔
if
elif
else
```

```
if      ✔
 if
 if
else
 if
```

```
if      ✘
else
    if
    elif
    elif
else
```

```
if      ✔
elif
```

# Looping

# Control Structures

- Structures that control how the program "flows" or operates, and in what order

- Sequence ✓  we've already seen these
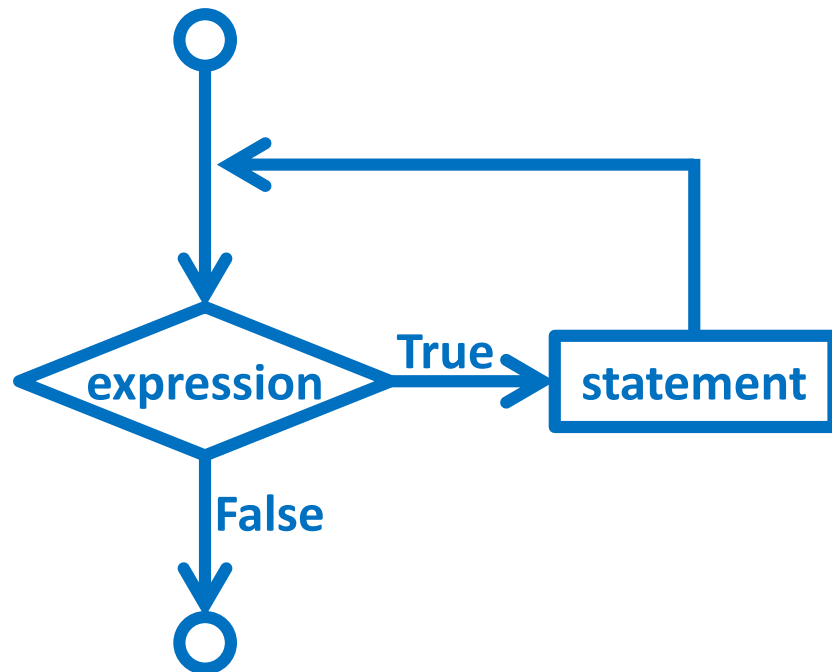
- Decision Making ✓

- Looping  what we're covering today

# Looping

- Doing something over and over (and over) again

- Used in combination with decision making
  - Otherwise we loop forever
    - This is called an "infinite loop"

- What are some real life examples?
  - Jumping rope
  - Walking up steps

# Looping

- We'll cover this in detail today

- It looks something like this…

# Looping

- Python has two kinds of loops, and they are used for two different purposes

- The **while** loop

  what we're covering today

  – Works for basically everything

- The **for** loop:

  – Best at *iterating* over a list

  – Best at counted iterations

# The **while** Loop

# "`while`" Loops

- `while <condition>:`
    `<body>`

- The **body** is a sequence of one or more statements <u>indented</u> under the heading

    – As long as the **condition** is `True`, the **body** will run (repeatedly if needed)

# How a `while` Loop Works

- The `while` loop requires a Boolean condition
  - That evaluates to either `True` or `False`


- If the condition is `True`:
  - Body of `while` loop is executed
  - Condition is checked again
- If the condition is `False`:
  - Body of `while` loop is skipped

# Parts of a **while** Loop

- Here's some example code... let's break it down

```
date = 0


while date < 1 or date > 28:
    date = int(input("Enter the day: "))


print("Today is February", date)
```

# Parts of a `while` Loop

- Here's some example code… let's break it down

initialize the variable the **while** loop will use for its decision

```
date = 0
```

the loop's Boolean condition (loop runs until this is **False**)

```
while date < 1 or date > 28:
    date = int(input("Enter the day: "))
```

the body of the loop (must change the value of the loop variable)

```
print("Today is February",
```
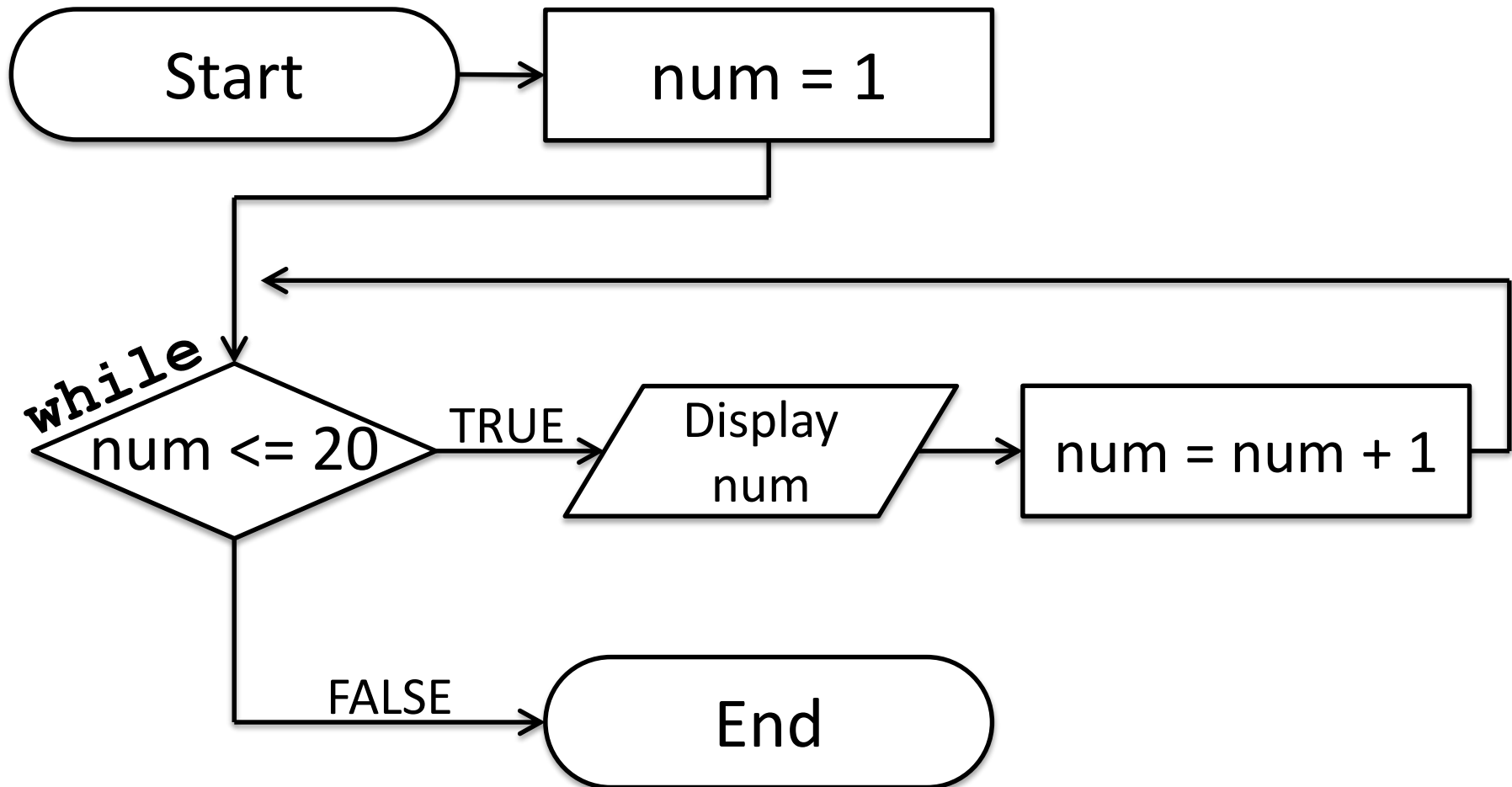
# Example **while** Loop

- We can use a **while** loop to count
  - Count from 1 up to and including 20

```
num = 1                  # we have to initialize num

while num <= 20:         # so that we can use it here
    print(num)
    num = num + 1        # don't forget to update
                         # the loop variable
```

# Example `while` Loop

Start → num = 1

**while**

num <= 20 — TRUE → Display num → num = num + 1

FALSE → End

# Infinite Loops and Other Problems

# Infinite Loops

- An *infinite loop* is a loop that will run forever
  - The conditional the loop is based on always evaluates to `True`, and never to `False`

- Why might this happen?
  - The loop variable is not updated
  - The loop variable is updated wrong
  - The loop conditional uses the wrong variable
  - The loop conditional checks the wrong thing

# Infinite Loop Example #1

- Why doesn't this loop end?  What will fix it?

```
age = 0
while age < 18:    # can't vote until 18
    print("You can't vote at age", age)

print("Now you can vote! Yay!")
```

# Infinite Loop Example #1

- Why doesn't this loop end?  What will fix it?

the loop variable (**age**) never changes, so the condition will never evaluate to **False**

```
age = 0
while age < 18:    # can't vote until 18
    print("You can't vote at age", age)


print("Now you can vote! Yay!")
```

# Infinite Loop Example #2

- Why doesn't this loop end?  What will fix it?

```python
while True:
    # ask user for name
    name = input("What is your name? ")

print("Hello", name + "!")
```

# Infinite Loop Example #2

- Why doesn't this loop end?  What will fix it?

> **True** will never evaluate to **False**, so the loop will never exit

```
while True:
    # ask user for name
    name = input("What is your name? ")

print("Hello", name + "!")
```

# Infinite Loop Example #3

- Why doesn't this loop end?  What will fix it?

```python
cookiesLeft = 50

while cookiesLeft > 0:
    # eat a cookie
    cookiesLeft = cookiesLeft + 1

print("No more cookies!")
```

# Infinite Loop Example #3

- Why doesn't this loop end?  What will fix it?

```
cookiesLeft = 50


while cookiesLeft > 0:

    # eat a cookie
    cookiesLeft = cookiesLeft + 1


print("No more cookies!")
```

the loop body is INCREASING the number of cookies, so we'll never reach zero!

# Infinite Loop Example #4

- Why doesn't this loop end?  What will fix it?

```python
grade = ""
name  = ""
while name != "Hrabowski":
    # get the user's grade
    grade = input("What is your grade? ")

print("You passed!")
```

# Infinite Loop Example #4

- Why doesn't this loop end?  What will fix it?

the loop conditional is checking the wrong thing! we also never change the name, so this will never end

```python
grade = ""
name  = ""
while name != "Hrabowski":
    # get the user's grade
    grade = input("What is your grade? ")

print("You passed!")
```

# Ending an Infinite Loop

- If you run a program that contains an infinite loop, it may seem like you've lost control of the terminal!

- To regain control, simply type **CTRL+C** to interrupt the infinite loop
  - **KeyboardInterrupt**

# Loop Body Not Being Reached

- A **while** loop's body may be skipped over entirely
  - If the Boolean condition is initially **False**

```
militaryTime = 1300


while (militaryTime < 1200):
    print("Good morning!")
    militaryTime = militaryTime + 100
```

# Practice with Decisions

# Loop Example #4 – Fixing It

- Let's update this to ask for the user's grade
  - An "A" or a "B" means that they passed

```
grade = ""
while   ...what goes here?
    # get the user's grade
    grade = input("What is your grade? ")

print("You passed!")
```

# Loop Example #4 – Truth Table

- Let's evaluate this expression

**`grade != "A" or grade != "B"`**

| grade | grade != "A" | grade != "B" | or |
|-------|--------------|--------------|-----|
| "A"   |              |              |     |
| "B"   |              |              |     |
| "C"   |              |              |     |

# Loop Example #4 – Truth Table

- Let's evaluate this expression

   **grade != "A" or grade != "B"**

| grade | grade != "A" | grade != "B" | or |
|-------|--------------|--------------|------|
| "A"   | False        | True         | True |
| "B"   | True         | False        | True |
| "C"   | True         | True         | True |

- This does not give us the answer we want
  - This just loops forever and ever (infinitely)

# Loop Example #4 – Truth Table

- Let's try it with an **and** instead of an **or**

  `grade != "A" and grade != "B"`

| grade | grade != "A" | grade != "B" | and |
|-------|--------------|--------------|-----|
| "A"   |              |              |     |
| "B"   |              |              |     |
| "C"   |              |              |     |

# Loop Example #4 – Truth Table

- Let's try it with an  **`and`**  instead of an  **`or`**

  **`grade != "A" and grade != "B"`**

| `grade` | `grade != "A"` | `grade != "B"` | `and` |
|---------|----------------|----------------|-------|
| `"A"`   | `False`        | `True`         | `False` |
| `"B"`   | `True`         | `False`        | `False` |
| `"C"`   | `True`         | `True`         | `True` |

- Now our program will behave how we want
  – You will sometimes have to stop and make a table!

# Loop Example #4 – Completed

- Let's update this to ask for the user's grade
  - An "A" or a "B" means that they passed

```python
grade = ""
while grade != "A" and grade != "B":
    # get the user's grade
    grade = input("What is your grade? ")

print("You passed!")
```

# Interactive **while** Loops

# When to Use `while` Loops

- `while` loops are very helpful when you want to get input from the user that meets certain specific conditions
  - Positive number
  - A non-empty string
  - A number within a certain range

# Example `while` Loop

- We can use a `while` loop to get correct input from the user by re-prompting them

```python
num  = 0                # we have to initialize num

while num <= 0:    # so that we can use it here
    num = int(input("Enter a positive number: "))

# while loop exits because num is positive
print("Thank you.  The number you chose is:", num)
```

# Updated Exercise: Nail Polish

- Update your nail polish "guessing game" to keep re-prompting the user until they guess the right number of bottles

- At each step, tell them whether their guess was high or low

- Exit the loop when they guess correctly

# Announcements

- HW 2 is out on Blackboard now
  - Complete the Academic Integrity Quiz to see it
  - Due by Friday (Feb 17th) at 8:59:59 PM

- Make sure to spell the dog breeds correctly!
  - Will make it much easier for your TA to grade

- Pre Lab 4 Quiz will come out Friday @ 10 AM
  - Must be <u>completed</u> by 9 AM Monday morning