

- Previous Lecture (and lab):
 - Variables & assignment
 - Built-in functions
 - Input & output
 - Good programming style (meaningful variable names; use comments)
- Today's Lecture:
 - Branching (conditional statements)

Announcements:

- Discussion this week in computer lab **Upson B7**
- Project I (P1) due Thurs, 9/3, at 11pm
- Pay attention to Academic Integrity
- You can see any TA for help, not just your discussion TA
- Matlab consultants at ACCEL Green Rm (Carpenter Hall 2nd fl. computing facility) 5-10pm Sunday to Thursday
- Just added CS1112? Tell your discussion TA to add you in CS1112 CMS (and tell CS1110 to drop your from their CMS)
- Answer "Week 1 survey" in CMS
- Piazza – "Q & A system" for all students in CS1112. Use it for clarification only—do not ask (answer) homework questions and do not give hints on homework. Will be monitored by TAs. Available later today.
- You must register your clicker using the link on course website

Quick review

- Variable
 - A named memory space to store a value
- Assignment operator: =
 - Let x be a variable that has a value. To give variable y the same value as x, which statement below should you write?
 $x = y$ or $y = x$
- Script (program)
 - A sequence of statements saved in an m-file
- ; (semi-colon)
 - Suppresses printing of the result of assignment statement

Lecture 3 3

Example

Modify the previous program to calculate the increase in surface area (mile²) given an increase in the radius of a sphere.

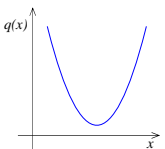
Note: 1 mile = 5280 feet

Lecture 2 4

- So far, all the statements in our scripts are executed in order
- We do not have a way to specify that some statements should be executed only under some condition
- We need a new language construct...

Lecture 3 5

Consider the quadratic function

$$q(x) = x^2 + bx + c$$


on the interval $[L, R]$:

- Is the function strictly increasing in $[L, R]$?
- Which is **smaller**, $q(L)$ or $q(R)$?
- What is the **minimum value** of $q(x)$ in $[L, R]$?

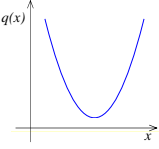
Lecture 3 6

■ What are the critical points?

- End points: $x = L, x = R$
- $\{ x \mid q'(x) = 0 \}$

$$q(x) = x^2 + bx + c$$

$$q'(x) = 2x + b$$

$$q'(x_c) = 0 \Rightarrow x_c = -\frac{b}{2}$$


Lecture 3 9

Problem 1

Write a code fragment that prints “yes” if $q(x)$ increases across the interval and “no” if it does not.

Lecture 3 10

```

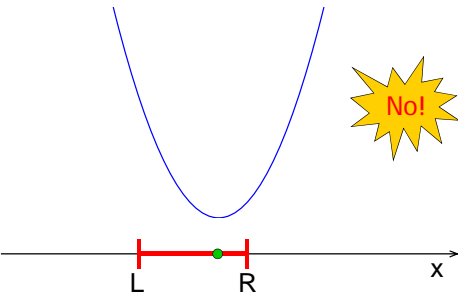
% Quadratic q(x) = x^2 + bx + c
b = input('Enter b: ');
c = input('Enter c: ');
L = input('Enter L: ');
R = input('Enter R: ');

% Determine whether q increases
% across [L,R]
xc = -b/2;
    
```

Lecture 3 11

Does $q(x)$ increase across $[L,R]$?

$q(x) = x^2 + bx + c$ • $x_c = -b/2$



Lecture 3 13

So what is the requirement?

```

% Determine whether q increases
% across [L,R]
xc = -b/2;

if _____
    fprintf('Yes\n')
else
    fprintf('No\n')
end
    
```

Relational Operators

- < Less than
- > Greater than
- <= Less than or equal to
- >= Greater than or equal to
- == Equal to
- ~= Not equal to


Lecture 3 16

So what is the requirement?

```

% Determine whether q increases
% across [L,R]
xc = -b/2;

if _____
    disp('Yes')
else
    disp('No')
end
    
```



Lecture 3 21

Problem 2

Write a code fragment that prints
 "qleft is smaller"
 if q(L) is smaller than q(R).
 If q(R) is smaller print
 "qright is smaller."

Lecture 3

22

Algorithm v0

Calculate q(L)
 Calculate q(R)
 If q(L) < q(R)
 print "qleft is smaller"
 Otherwise
 print "qright is smaller"

Lecture 3

23

Algorithm v0.1

Calculate x_c
 If distance $\overline{x_c L}$ is smaller than distance $\overline{x_c R}$
 print "qleft is smaller"
 Otherwise
 print "qright is smaller"

Lecture 3

24

Do these two fragments do the same thing?

```
% given x, y
if x>y
    disp('alpha')
else
    disp('beta')
end
```

```
% given x, y
if y>x
    disp('beta')
else
    disp('alpha')
end
```

A: yes

B: no

Lecture 3

25

```
% Which is smaller, q(L) or q(R)?

xc= -b/2; % x at center
if (abs(xc-L) == abs(xc-R))
    disp('qleft and qright are equal')
elseif (abs(xc-L) < abs(xc-R))
    disp('qleft is smaller')
else
    disp('qright is smaller')
end
```

Lecture 3

28

```
% Which is smaller, q(L) or q(R)?

qL= L*L + b*L + c; % q(L)
qR= R*R + b*R + c; % q(R)
if (qL == qR)
    disp('qleft and qright are equal')
    fprintf('q value is %f\n', qL)
elseif (qL < qR)
    disp('qleft is smaller')
else
    disp('qright is smaller')
end
```

Lecture 3

30

Consider the quadratic function

$$q(x) = x^2 + bx + c$$

on the interval $[L, R]$:

What if you only want to know if $q(L)$ is close to $q(R)$?

Lecture 3 31

```
% Is q(L) close to q(R)?

tol= 1e-4; % tolerance
qL= L*L + b*L + c
qR= R*R + b*R + c
if (abs(qL-qR) < tol)
    disp('qleft and qright similar')
end
```

Name an important parameter and define it with a comment!

Lecture 3 32

Do these two fragments do the same thing?

<pre><code>% given x, y if x>y disp('alpha') else disp('beta') end</code></pre>	<pre><code>% given x, y if x>y disp('alpha') end if y>=x disp('beta') end</code></pre>
<div style="display: inline-block; border: 1px solid black; padding: 2px; margin: 5px;">A: yes</div> <div style="display: inline-block; border: 1px solid black; padding: 2px; margin: 5px; margin-left: 20px;">B: no</div>	

Lecture 3 33

Simple **if** construct

```
if boolean expression
    statements to execute if expression is true
else
    statements to execute if expression is false
end
```

Lecture 3 34

Even simpler **if** construct

```
if boolean expression
    statements to execute if expression is true
end
```

Lecture 3 35

The **if** construct

```
if boolean expression1
    statements to execute if expression1 is true
elseif boolean expression2
    statements to execute if expression1 is false
    but expression2 is true
:
else
    statements to execute if all previous conditions
    are false
end
```

Can have any number of elseif branches but at most one else branch

Lecture 3 36

Things to know about the `if` construct

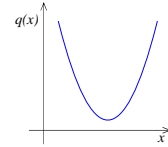
- _____ branch of statements is executed
- There can be _____ `elseif` clauses
- There can be _____ `else` clause
- The `else` clause _____ in the construct
- The `else` clause _____ (boolean expression)

Lecture 3

37

Consider the quadratic function

$$q(x) = x^2 + bx + c$$



on the interval $[L, R]$:

- Is the function strictly increasing in $[L, R]$?
- Which is smaller, $q(L)$ or $q(R)$?
- What is the minimum value of $q(x)$ in $[L, R]$?

Lecture 3

40

Modified Problem 3

Write a code fragment that prints “yes” if `xc` is in the interval and “no” if it is not.

Lecture 3

41

So what is the requirement?

```

% Determine whether xc is in
% [L,R]
xc = -b/2;

if _____

    disp('Yes')
else
    disp('No')
end
    
```

Lecture 3

45

The value of a boolean expression is either `true` or `false`.

$$(L \leq xc) \ \&\& \ (xc \leq R)$$

This (compound) boolean expression is made up of two (simple) boolean expressions. Each has a value that is either `true` or `false`.

Connect boolean expressions by **boolean** operators:

and	or	not
<code>&&</code>	<code> </code>	<code>~</code>

Lecture 3

47

Logical operators

- `&&` logical **and**: Are both conditions true?
E.g., we ask “is $L \leq x_c$ and $x_c \leq R$?”
In our code: `L <= xc && xc <= R`
- `||` logical **or**: Is at least one condition true?
E.g., we can ask if x_c is outside of $[L, R]$,
i.e., “is $x_c \leq L$ or $R \leq x_c$?”
In code: `xc < L || R < xc`
- `~` logical **not**: Negation
E.g., we can ask if x_c is **not outside** $[L, R]$.
In code: `~(xc < L || R < xc)`

Lecture 3

51