

CSE 100: Introduction to Computer Systems

Tanvir Ahmed Khan

Department of Computer Science and Engineering
Bangladesh University of Engineering and Technology.

August 23, 2014

Today's Topic

তুমি বরণা হলে হবো আমি সুনীল
তুমি আকাশ হলে হবো শংখচীল

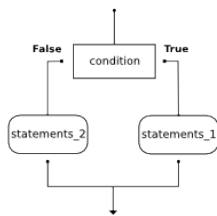
Today's Topic

তুমি বরুণা হলে হবো আমি সুনীল
তুমি আকাশ হলে হবো শংখচীল

যদি তুমি == বরুণা
আমি = সুনীল
যদি তুমি == আকাশ
আমি = শংখচীল

Today's Topic

Conditionals



Boolean Values

- ▶ True
- ▶ False

Boolean Values

- ▶ True
- ▶ False

```
>>> print(True)
True
>>> print(False)
False
```

Boolean Values

- ▶ True
- ▶ False

```
>>> print(True)
True
>>> print(False)
False
>>> print(type(False))
```

Boolean Values

- ▶ True
- ▶ False

```
>>> print(True)
True
>>> print(False)
False
>>> print(type(False))
<class 'bool'>
```


Boolean Values

- ▶ True
- ▶ False

```
>>> print(True)
True
>>> print(False)
False
>>> print(type(False))
<class 'bool'>
>>> print(type('True'))
```

Boolean Values

- ▶ True
- ▶ False

```
>>> print(True)
True
>>> print(False)
False
>>> print(type(False))
<class 'bool'>
>>> print(type('True'))
<class 'str'>
```

Boolean Values

- ▶ True
- ▶ False

```
>>> print(True)
True
>>> print(False)
False
>>> print(type(False))
<class 'bool'>
>>> print(type('True'))
<class 'str'>
>>> print(type(True))
<class 'bool'>
```

Boolean Expression

Boolean Expression

- ▶ results into a boolean value

Boolean Expression

- ▶ results into a boolean value

```
>>> print(3 == 3)
```

```
True
```

```
>>> print(3 == 1)
```

```
False
```

Boolean Expression

- ▶ results into a boolean value

```
>>> print(3 == 3)
```

```
True
```

```
>>> print(3 == 1)
```

```
False
```

```
>>> print('CSE 100' ==  
'CSE 100')
```

```
True
```

Relational Operators

<code>==</code>	<code>x == y</code>	x is equal to y
<code>!=</code>	<code>x != y</code>	x is not equal to y
<code>></code>	<code>x > y</code>	x is greater than y
<code><</code>	<code>x < y</code>	x is less than y
<code>>=</code>	<code>x >= y</code>	x is greater than or equal to y
<code><=</code>	<code>x <= y</code>	x is less than or equal to y

Relational Operators

<code>==</code>	<code>x == y</code>	x is equal to y
<code>!=</code>	<code>x != y</code>	x is not equal to y
<code>></code>	<code>x > y</code>	x is greater than y
<code><</code>	<code>x < y</code>	x is less than y
<code>>=</code>	<code>x >= y</code>	x is greater than or equal to y
<code><=</code>	<code>x <= y</code>	x is less than or equal to y

*** `==` is different from `=`

Relational Operators

<code>==</code>	<code>x == y</code>	x is equal to y
<code>!=</code>	<code>x != y</code>	x is not equal to y
<code>></code>	<code>x > y</code>	x is greater than y
<code><</code>	<code>x < y</code>	x is less than y
<code>>=</code>	<code>x >= y</code>	x is greater than or equal to y
<code><=</code>	<code>x <= y</code>	x is less than or equal to y

*** `==` is different from `=`

*** no such thing as `=<` or `=>`

Quick Quiz

Valid Boolean Expression?

1. True
2. `2 == 1`
3. `"False"`
4. `21 + 60`
5. `'Python' == 'python'`

Quick Quiz

Valid Boolean Expression?

1. **True**
2. **2 == 1**
3. "False"
4. 21 + 60
5. **'Python' == 'python'**

Logical Operators

- ▶ `and`
- ▶ `or`
- ▶ `not`

Logical Operators

- ▶ `and`
- ▶ `or`
- ▶ `not`

```
>>> x = 3
>>> print(x > 1 and x <
21)
```

Logical Operators

- ▶ `and`
- ▶ `or`
- ▶ `not`

```
>>> x = 3
>>> print(x > 1 and x <
21)
True
```

Logical Operators

- ▶ `and`
- ▶ `or`
- ▶ `not`

```
>>> x = 3
>>> print(x > 1 and x <
21)
True
>>> n = 25
>>> print(n % 2 == 0 or
not n % 5 == 0)
```


Logical Operators

- ▶ `and`
- ▶ `or`
- ▶ `not`

```
>>> x = 3
>>> print(x > 1 and x <
21)
True
>>> n = 25
>>> print(n % 2 == 0 or
not n % 5 == 0)
False
```

Quick Test

Your student id is x , expression to check-

Quick Test

Your student id is x , expression to check-

- ▶ you are in section B.

Quick Test

Your student id is x , expression to check-

- ▶ you are in section B.
- ▶ you are in lab section B2

Quick Test

Your student id is x , expression to check-

- ▶ you are in section B.
- ▶ you are in lab section B2
- ▶ you are not in lab section B1

Precedence & Associativity of Operators

Operators	Category	Associativity
()	Parentheses	-
**	Exponent	R →L
*, /, //, %	Multiplication	L →R
+, -	Addition	L →R
==, !=, <=, >=, >, <	Relational	-
not	Logical	L →R
and	Logical	L →R
or	Logical	L →R

Quick Test

$$5+18//3*2**2**3-1 = ?$$

1. 388
2. 1540
3. 4

Quick Test

$$5+18//3*2**2**3-1$$

$$5+18//3*2**2**3-1 = ?$$

1. 388
2. 1540
3. 4

Quick Test

$$5+18//3*2**2**3-1$$

$$5+18//3*2**2**3-1$$

$$5+18//3*2**8-1$$

$$5+18//3*2**2**3-1 = ?$$

1. 388
2. 1540
3. 4

Quick Test

$$5+18//3*2**2**3-1 = ?$$

1. 388
2. 1540
3. 4

$$5+18//3*2**2**3-1$$

$$5+18//3*2**2**3-1$$

$$5+18//3*2**8-1$$

$$5+18//3*2**8-1$$

$$5+18//3*256-1$$

Quick Test

$$5+18//3*2**2**3-1 = ?$$

1. 388
2. 1540
3. 4

$$5+18//3*2**2**3-1$$

$$5+18//3*2**2**3-1$$

$$5+18//3*2**8-1$$

$$5+18//3*2**8-1$$

$$5+18//3*256-1$$

$$5+18//3*256-1$$

$$5+6*256-1$$

Quick Test

$$5+18//3*2**2**3-1 = ?$$

1. 388
2. 1540
3. 4

$$5+18//3*2**2**3-1$$

$$5+18//3*2**2**3-1$$

$$5+18//3*2**8-1$$

$$5+18//3*2**8-1$$

$$5+18//3*256-1$$

$$5+18//3*256-1$$

$$5+6*256-1$$

$$5+6*256-1$$

$$5+1536-1$$

Quick Test

$$5+18//3*2**2**3-1 = ?$$

1. 388
2. 1540
3. 4

$$5+18//3*2**2**3-1$$

$$5+18//3*2**2**3-1$$

$$5+18//3*2**8-1$$

$$5+18//3*2**8-1$$

$$5+18//3*256-1$$

$$5+18//3*256-1$$

$$5+6*256-1$$

$$5+6*256-1$$

$$5+1536-1$$

$$5+1536-1$$

$$1541-1$$

Quick Test

$$5+18//3*2**2**3-1 = ?$$

1. 388
2. 1540
3. 4

$$5+18//3*2**2**3-1$$

$$5+18//3*2**2**3-1$$

$$5+18//3*2**8-1$$

$$5+18//3*2**8-1$$

$$5+18//3*256-1$$

$$5+18//3*256-1$$

$$5+6*256-1$$

$$5+6*256-1$$

$$5+1536-1$$

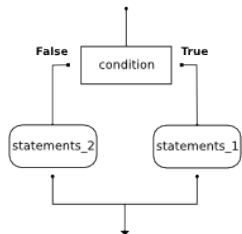
$$5+1536-1$$

$$1541-1$$

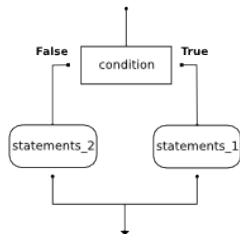
$$1540$$

At Last!

Conditional Execution



Conditional Execution



```
if BOOLEAN EXPRESSION:  
    STATEMENTS_1      # executed if condition evaluates to True  
else:  
    STATEMENTS_2      # executed if condition evaluates to False
```

Conditional Execution

```
6 id = 100
7
8 if id % 2 == 0:
9     section = 'B'
10    print("Your section is", section)
11    pass
12 else:
13    section = 'A'|
14    print("Your section is", section)
15    pass
```

Conditional Execution

```
6 id = 100
7
8 if id % 2 == 0:
9     section = 'B'
10    print("Your section is", section)
11    pass
12 else:
13    section = 'A'
14    print("Your section is", section)
15    pass
```

Your section is B

Conditional Execution

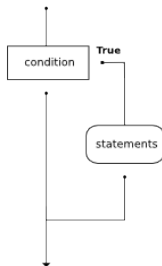
```
6 id = 101|
7
8 if id % 2 == 0:
9     section = 'B'
10    print("Your section is", section)
11    pass
12 else:
13    section = 'A'
14    print("Your section is", section)
15    pass
--
```

Conditional Execution

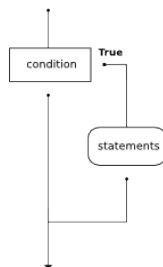
```
6 id = 101|
7
8 if id % 2 == 0:
9     section = 'B'
10    print("Your section is", section)
11    pass
12 else:
13    section = 'A'
14    print("Your section is", section)
15    pass
--
```

Your section is A

Conditional Execution

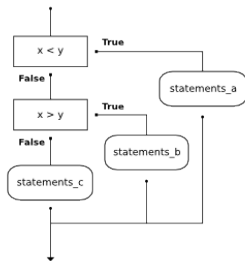


Conditional Execution

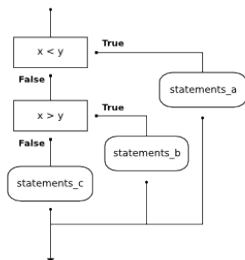


```
17 if id > 0 and id < 121:
18     print("This is a valid id.")
19     pass
20 |
```

Chained Conditionals



Chained Conditionals



```
20  
21 if x < y:  
22     print("x is less than y")  
23 elif x > y:  
24     print("x is greater than y")  
25 else:  
26     print("x and y must be equal")  
27
```

Nested Conditionals

```
32 if std_id % 2 == 0:
33     if std_id < 61:
34         print("Sessional, B1")
35     else:
36         print("Sessional, B2")
37 else:
38     if std_id < 61:
39         print("Sessional, A1")
40     else:
41         print("Sessional, A2")
42
```

Practice Problems

- ▶ Find only the real solutions of the quadratic equation.
- ▶ Test whether a given year is leap year or not.