Certified Python Programmer - PCEP

Montgomery College

Workforce Development & Continuing Education

Information Technology Institute

Lesson Objectives

Lesson 4

Strings

String Methods

Formatted Strings

More Numbers


Strings

Strings

Lesson 4

address = “123 Sesame Street”

state = ‘MD’

zipcode = “12345”

phone = ‘301-555-1234’

Strings

fname = “Cookie”

String Operations

fname = “Cookie”

fname → C o o k i e

lname = “Monster”

lname → M o n s t e r

fname + lname → C o o k i e M o n s t e r

fname * 2 → C o o k i e C o o k i e

STRING methods

string.method1()

string.method2(*parameter*)

string.method3(*param1, param2…*)

STRING methods

message = “ learn to code in python ”

message.**title()**	“ Learn To Code In Python “
message.**upper()**	“ LEARN TO CODE IN PYTHON “
message.**strip()**	“learn to code in python”
message.**count(“to”)**	1
message.**replace(“ “, “*”)**	“**learn*to*code*in*python**”

Built-In Function: len()

Lesson 4

len(arg)

len(“Hello”)
len(“”)
len(“99.99”)

Classwork

- Do **Exercise 1 in Module4 Class Demo.ipynb**

- ***The result should look something like this:***

```
beautiful is better than ugly.
```


String Format

STRING format()

{ }	place holder
:,	commas in thousands place
:f	decimal points – default is 6
:%	percentage

message = “I love {name}”

STRING format()

message.format(name = “Python”)

“I love {0}”.format(“Python”)
“I love {}”. format(“Python”)

“I love Python”

STRING format()

“Price is ${ **:f** }”**.format(** 9.99 **)**
“Price is $9.990000”

“Price is ${ **:.2f** }”**.format(** 9.99 **)**
“Price is $9.99”

“Price is ${ **:.1f** }”**.format(** 9.99 **)**
“Price is $10.0”

STRING format()

“Rate is { **:%** }”**.format(** 0.0516 **)**
“Rate is 5.160000%”

“Rate is { **:.2%** }”**.format(** 0.0516 **)**
“Rate is 5.16%”

“Rate is ${ **:.1%** }”**.format(** 0.0516 **)**
“Rate is 5.2%”

Break

Take a 5-minute break

Formatted String – f string

Lesson 4

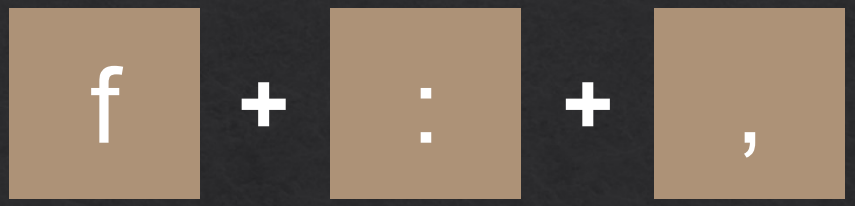
f

f “Hello, {user}”

Hello, Elmo

Formatted String – f string

Lesson 4



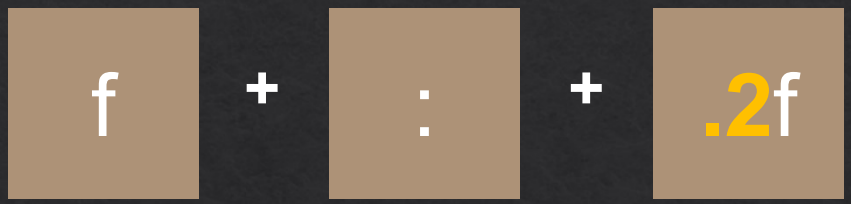
show commas in thousands place

```
amount = 1234567.89123
print( f “Amount is ${ amount:, }” )
```

Amount is $1,234,567.89123

Formatted String – f string

Lesson 4



show **2** decimal places

```
amount = 1234567.89123
print( f “Amount is ${ amount:.2f }” )
```

Amount is $1234567.89

Formatted String – f string

Lesson 4

f + : + .3%

show 3 decimal places in percentage (%)

```
rate = 0.0312
print( f “Rate is { rate:.3% }” )
```

Rate is 3.120%

Work on Exercise 2 in Module4 Class Demo: IPO & code

Classwork


More Numbers

More Numbers

Lesson 4

- Scientific Numbers
- Octal Numbers
- Hexadecimal Numbers

Scientific Numbers

Lesson 4

$314 = 3.14 \times 10^2$

3.14e2

$123{,}000 = 1.23 \times 10^5$

1.23e5

$75{,}000{,}000 = 7.5 \times 10^7$

7.5e7

Scientific Numbers

Lesson 4

```
print( f " {314 :e } ")          3.140000e+2

print( f " {75000000 :.1e } " )  7.5e+7

print( f " {75000000 :.2e } " )  7.50e+7
```

Scientific Numbers

Lesson 4

```
print( f " {3.14e2 :f } ")      314.000000

print( f " {7.5e7 :.0f } ")     75000000

print( f " {7.5e7 :,.0f } ")    75,000,000
```

Scientific Numbers

More Basics

```
"{ :,.2f } ".format( 5432.1234 )
```

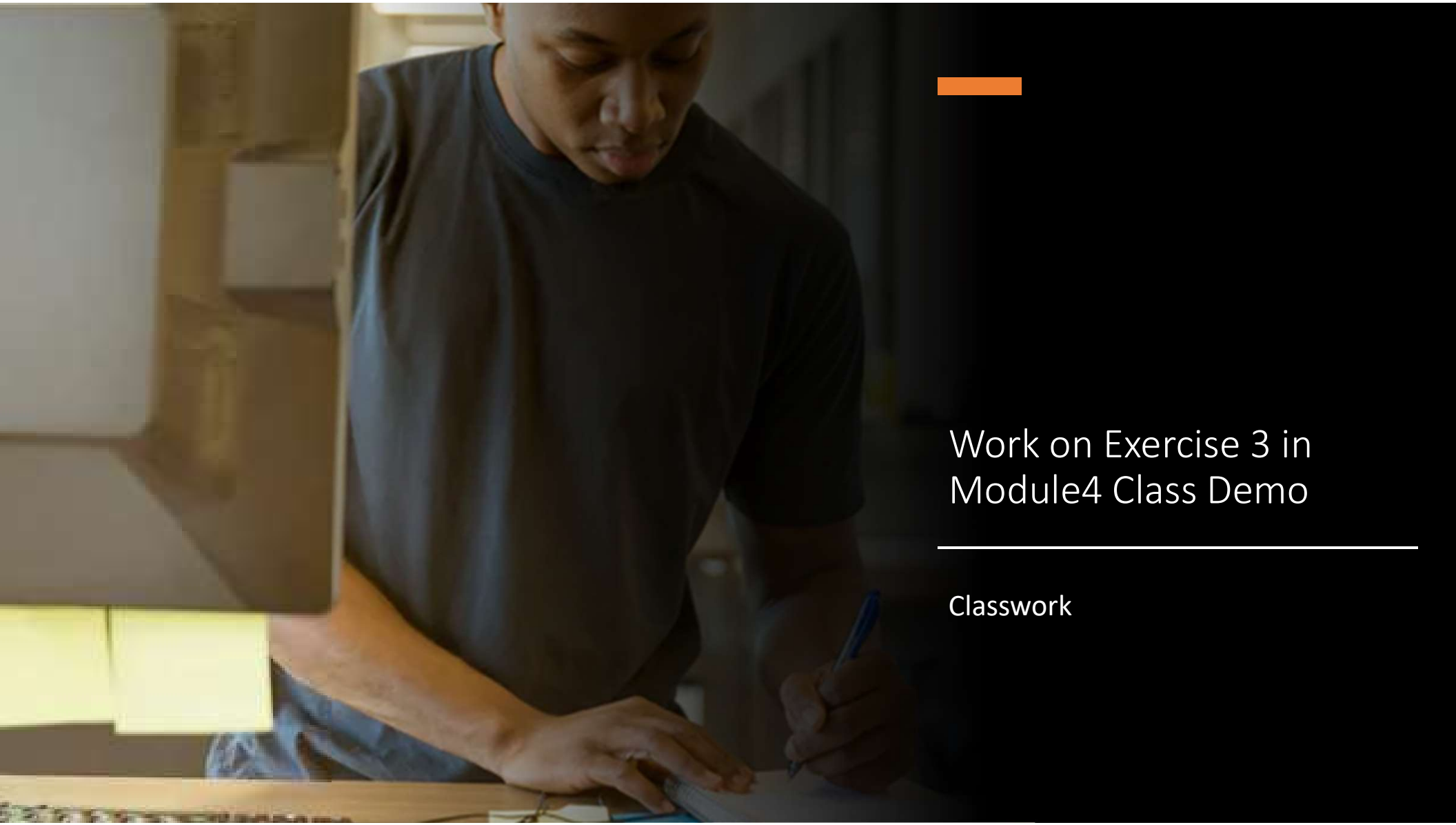
5,432.12

```
print( "{ :f }".format( 3.14e+2 ))
```

314.000000

```
print("{ :.3e }".format(5432.1234))
```

5.432e+3



Work on Exercise 3 in Module4 Class Demo

Classwork

Break

Take a 5-minute break

Binary Numbers

Lesson 4

11

$1 * 10^1 + 1 * 10^0 = 11$

$1 * 2^1 + 1 * 2^{*0} = 3$

Binary Numbers

Lesson 4

0b10011011

$1 * 2^7 = 1 * (2 * 2 * 2 * 2 * 2 * 2 * 2) = 1 * 128$

$0 * 2^6 = 0 * (2 * 2 * 2 * 2 * 2 * 2) = 0 * 64$

$0 * 2^5 = 0 * (2 * 2 * 2 * 2 * 2) = 0 * 32$

$1 * 2^4 = 1 * (2 * 2 * 2 * 2) = 1* 16$

$1 * 2^3 = 1 * (2 * 2 * 2) = 1 * 8$

$0 * 2^2 = 0 * (2 * 2) = 0 * 4$

$1 * 2^1 = 1 * (2) = 1 * 2$

$1 * 2^0 = 1 * (1) = 1 * 1$

$128 + 16 + 8 + 2 + 1 = 155$

Binary Numbers

Lesson 4

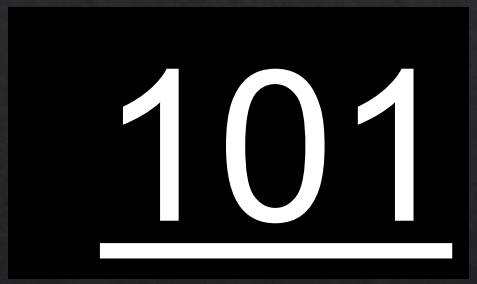
155

155 // 2 = 77;	155 % 2 = 1
77 // 2 = 38;	77 % 2 = 1
38 // 2 = 19;	38 % 2 = 0
19 // 2 = 9;	19 % 2 = 1
9 // 2 = 4;	9 % 2 = 1
4 // 2 = 2;	4 % 2 = 0
2 // 2 = 1;	2 % 2 = 0
1 // 2 = 0;	1 % 2 = 1

0b10011011

Octal Numbers

Lesson 4



Octal Numbers

Lesson 4

0o1234567

$0 * 8^7 = 0 * (8 * 8 * 8 * 8 * 8 * 8 * 8) = 0 * 2097152$

$1 * 8^6 = 1 * (8 * 8 * 8 * 8 * 8 * 8) = 1 * 262144$

$2 * 8^5 = 2 * (8 * 8 * 8 * 8 * 8) = 2 * 32768$

$3 * 8^4 = 3 * (8 * 8 * 8 * 8) = 3* 4096$

$4 * 8^3 = 4 * (8 * 8 * 8) = 4 * 512$

$5 * 8^2 = 5 * (8 * 8) = 5 * 64$

$6 * 8^1 = 6 * (8) = 6 * 8$

$7 * 8^0 = 7 * (1) = 7 * 1$

342391

Octal Numbers

Lesson 4

342391 = 0o1234567

342391 // 8 = 42798; 342391 % 8 = 7

42798 // 8 = 5349; 42798 % 8 = 6

5349 // 8 = 668; 5349 % 8 = 5

668 // 8 = 83; 668 % 8 = 4

83 // 8 = 10; 83 % 8 = 3

10 // 8 = 1; 10 % 8 = 2

1 // 8 = 0; 1 % 8 = 1

01234567

Hexadecimal Numbers

Lesson 4

Hexadecimal Numbers

Lesson 4

0x20b **abcdef**

2: $2 * 16^2 = 2 * (16 * 16) = 2 * 256 = 512$

0: $0 * 16^1 = 0 * (16) = 0$

B: $11 * 16^0 = 11 * (1) = 11$

$= 523$

Hexadecimal Numbers

Lesson 4

523 = 0x20b

523 // 16 = 32; 523 % 16 = 11

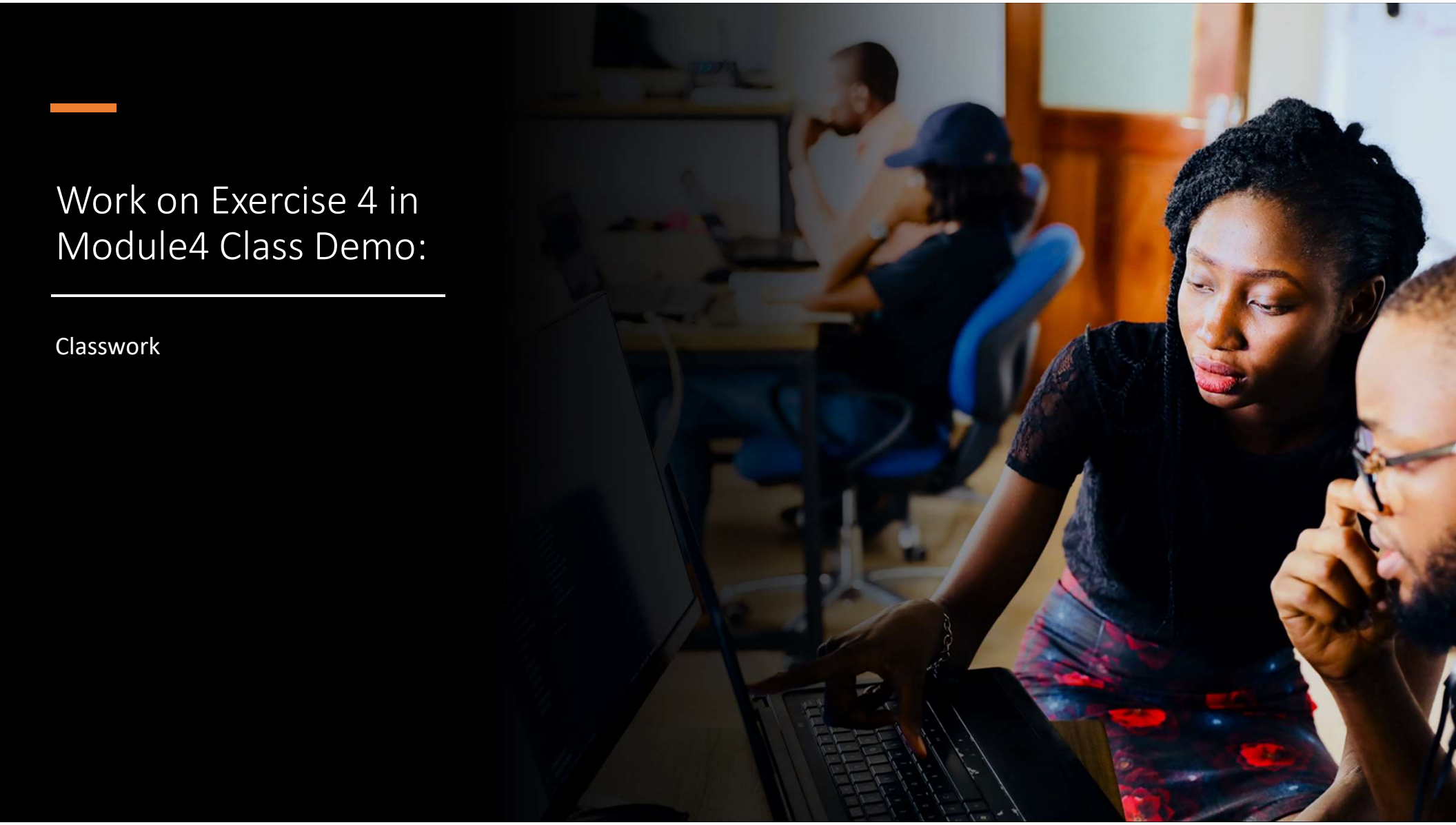
32 // 16 = 2; 32 % 16 = 0

2 // 16 = 0; 2 % 16 = 2

20b

Work on Exercise 4 in Module4 Class Demo:

Classwork



Data Type Casting

Lesson 4

```
bin(155)          "0b10011011"
oct(342391)       "0o1234567"
hex(523)          "0x20b"

int( 0b10011011 )     155
int( 0o1234567 )      342391
int( 0x20b )          523

int( "1011", 2 )      11
int( "1011", 8 )      521
int( "1011", 16 )     4113
```


Questions?

Homework

- Download Module4 Exercise from Canvas
- Change the file name to include your last name
- Submit your work to Canvas
- Watch recordings in for review


Thank you!