

(Monty) Python Graphics

Mr. Neat

Comparison

	Java	C++	Python
source code name	.java	.cpp	.py
Object Oriented?	required	optional	optional
output	<code>new Text(" ", , ,)</code> <code>System.out.print()</code>	<code>cout <<</code>	<code>print(" blah, blah")</code>
functions/ methods	methods (<code>bob.hide()</code>)	functions (<code>gotoxy(4,3)</code>)	functions (<code>str(5)</code>)
input	skipped this (complicated)	<code>cin>></code>	<code>input()</code>
casting <code>int()</code>	<code>int(double)</code>	<code>int(double)</code>	<code>int(string)</code>
if - else	<code>if(boolean)</code> { } <code>else</code> { }	<code>if(boolean)</code> { } <code>else</code> { }	brackets replaced by indenting!

Comparison

Java

C++

Python

for loops

`for(int i=0 ;i<10 ; i++)`

exact same as Java

different

graphics

used `ObjectDraw`

difficult

`graphics.py`

Graphics

- We will use a graphics package
- For now, include the graphics.py file in the same file as your .py file
- The following lab sequence will be very similar to the Java objectDraw sequence
- This folder contains a file (graphics.pdf) that has the documentation for this package –
READ IT

Graphics

- The sample.py file contained in this folder has a simple example on how to make a shape in a window
- Geometric shapes must be drawn in a window
- The window is created using the GraphWin class
- Comments in sample.py explain what each command does

Lab 10

- Experiment with the provide code in the .py file
 - Change the size of the circle
 - Change the center of the circle
 - Change the shape to a rectangle
 - Change its color