|  |  |
| --- | --- |
| **Date Assigned: 1/5/16** | **Date Due: 1/7/16** |
| **Unit:** Languages | **Turn In List:** **1. Terms** |
| *“I will be able to declare the language of focus for Semester 2 .”* | |

**Computer Programming Languages: An in-depth analysis**

**Content Objectives:** Students will research each of the five languages acceptable for the 2A computer programming state CTE certification. The following [Wiki article](http://en.wikipedia.org/wiki/Comparison_of_programming_languages) may help in your search. [Language popularity article](http://en.wikipedia.org/wiki/Measuring_programming_language_popularity).

|  |
| --- |
| **Starter Activity** |
| Write a class that will run in Processing. You may choose from the following list of class names: Human, Cat, Dog, Spaceship, Soldier or Planet. The class must contain a name, at least 2 class variables, 1 constructor, a display function and at least one action function. Paste code below: void setup() {  //set the size of the window  size(1500, 950);  }  void draw() {  //Draw a white background  background (0);  zoog(mouseX, mouseY);  //set CENTER mode  }  void zoog(int x, int y) {  rectMode(CENTER);  //Jack  fill(255);  text("JACK", x+35, y-30, 100, 100);  //Draw Zoog's body  stroke(0);  fill(150);  rect(x, y, 20, 100);  //Draw zoog's head  stroke(0);  fill(255);  ellipse(x, y-30, 60, 60);  //Draw zoog's nose  fill(0);  ellipse(x-5, y-20, 5, 5);  ellipse(x+5, y-20, 5, 5);  //Draw Zoog's eyes  fill(0);  ellipse(x-19, y-30, 16, 32);  ellipse(x+19, y-30, 16, 32);  //Draw Zoog's legs  fill(255);  stroke(255);  line(x-10, y+50, x-20, y+60);  line(x+10, y+50, x+20, y+60);  //Draw Zoog's Arms  fill(255);  stroke(255);  line(x-10, y+15, x-25, y+20);  line(x+10, y+15, x+25, y+20);  //moon  fill(#8B8A8A);  ellipse(750, 2000, 3000, 3000);  //sun  fill(random(0, 255), random(0, 255), random(0, 255));  ellipse(1250, 200, 200, 200);    //sun's rays  fill(random(0, 255), random(0, 255), random(0, 255));  line(1150,40,1190,90);  //craters  fill(0);  ellipse(200, 700, 100, 100);  ellipse(500, 834, 100, 100);  ellipse(800, 750, 100, 100);  ellipse(1200, 750, 100, 100);  ellipse(400, 700, 100, 100);  ellipse(600, 600, 100, 100);  ellipse(1050, 650, 100, 100);  //other planets  fill(#CB8C25);  ellipse(1100, 200, 40, 40);  fill(#EDD0A0);  ellipse(1000, 200, 60, 60);  fill(#232FFA);  ellipse(850,200,80,80);  } |

|  |  |
| --- | --- |
| **Key Terms: (lookup each language and write a short description of each)** | |
| **C++** | **C++** ([pronounced](https://en.wikipedia.org/wiki/English_alphabet#Letter_names) as *cee plus plus*, [/ˈsiː plus plus/](https://en.wikipedia.org/wiki/Help:IPA_for_English)) is a general-purpose [programming language](https://en.wikipedia.org/wiki/Programming_language). It has [imperative](https://en.wikipedia.org/wiki/Imperative_programming), [object-oriented](https://en.wikipedia.org/wiki/Object-oriented_programming) and [generic](https://en.wikipedia.org/wiki/Generic_programming) programming features, while also providing facilities for low-level memory manipulation. |
| **C#** | **C#** [note 2] (pronounced as see sharp) is a multi-paradigm programming language encompassing strong typing, imperative, declarative, functional, generic, object-oriented (class-based), and component-oriented programming disciplines. |
| **Java** | Java is a programming language expressly designed for use in the [distributed](http://searchcio-midmarket.techtarget.com/definition/distributed) environment of the Internet. It was designed to have the "look and feel" of the [C++](http://searchsqlserver.techtarget.com/definition/C) language, but it is simpler to use than C++ and enforces an [object-oriented programming](http://searchsoa.techtarget.com/definition/object-oriented-programming) model. |
| **Python** | a high-level general-purpose programming language. |
| **Visual Basic** | a programming environment from Microsoft in which a programmer uses a graphical user interface (GUI) to choose and modify preselected sections of code written in the **BASIC**programming language. |
| Type Safety | the extent to which a programming language discourages or prevents **type** errors. |
| Interpreted | An interpreted program, sometimes called a *script* , is a program whose instructions are actually a logically sequenced series of operating system commands, handled one at a time by a [command interpreter](http://searchwinit.techtarget.com/definition/command-interpreter) . |
| Procedural | Same as [routine](http://www.webopedia.com/TERM/R/routine.html), subroutine, and [function](http://www.webopedia.com/TERM/F/function.html). A procedure is a section of a [program](http://www.webopedia.com/TERM/P/program.html) that performs a specific task. |
| Compiled | A compiler is a **computer** program (or a set of programs) that transforms source code written in a programming language (the source language) into another **computer** language (the target language), with the latter often having a binary form known as object code. |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **C++** | **C#** | **Java** | **Python** | **Visual Basic** |
| Intended Use | games, office applications, graphics and video editors, and operating systems. | C# is intended to be a simple, modern, general-purpose, object-oriented programming language. | **Java** is a general-purpose [computer programming language](https://en.wikipedia.org/wiki/Programming_language) that is [concurrent](https://en.wikipedia.org/wiki/Concurrent_computing), [class-based](https://en.wikipedia.org/wiki/Class-based_programming), [object-oriented](https://en.wikipedia.org/wiki/Object-oriented_programming),[[12]](https://en.wikipedia.org/wiki/Java_(programming_language)#cite_note-FOOTNOTEGoslingJoySteeleBracha20141-12) and specifically designed to have as few implementation dependencies as possible. | **Python** is a widely used [general-purpose](https://en.wikipedia.org/wiki/General-purpose_programming_language), [high-level programming language](https://en.wikipedia.org/wiki/High-level_programming_language). | Visual Basic was designed to accommodate a steep [learning curve](https://en.wikipedia.org/wiki/Learning_curve). |
| Strongly Typed | no | yes | yes | yes | both |
| OS’s | Windows, os, linux | windows | Windows, os, linux | Windows, os, linux | windows |
| Industry | microsoft | microsoft | Oracle | Oracle | microsoft |
| **Atoms or Bits** | bits | bits | bits | bits | atoms |
| Current Version | 11 | 5.0 | 8 | 3.4.3rc1 | 10 |
| Official Standard | 11 | 5.0 | 7 | 3.4.2 | 10 |

|  |
| --- |
| **History and Background of the Language you are interested in:** |
| You may work in pairs for this portion but you need to submit your own file to Canvas. Give the When’s, Who’s, Why’s, Where’s, How’s and worldwide popularity pulse applicable for the language you are considering. (Note, this is NOT your final decision.)  Java |

|  |
| --- |
| **Assignment:** |
| Rewrite Class from Starter:  Find the official standard website or simply do a google search for your language and “class” or “object” and do your best to re-write the class from starter in the new language (code not require to build or compile.) //set the size of the window  size(1500, 950);  }  void draw() {  //Draw a white background  background (0);  zoog(mouseX, mouseY);  //set CENTER mode  }  void zoog(int x, int y) {  rectMode(CENTER);  //Jack  fill(255);  text("JACK", x+35, y-30, 100, 100);  //Draw Zoog's body  stroke(0);  fill(150);  rect(x, y, 20, 100);  //Draw zoog's head  stroke(0);  fill(255);  ellipse(x, y-30, 60, 60);  //Draw zoog's nose  fill(0);  ellipse(x-5, y-20, 5, 5);  ellipse(x+5, y-20, 5, 5);  //Draw Zoog's eyes  fill(0);  ellipse(x-19, y-30, 16, 32);  ellipse(x+19, y-30, 16, 32);  //Draw Zoog's legs  fill(255);  stroke(255);  line(x-10, y+50, x-20, y+60);  line(x+10, y+50, x+20, y+60);  //Draw Zoog's Arms  fill(255);  stroke(255);  line(x-10, y+15, x-25, y+20);  line(x+10, y+15, x+25, y+20);  //moon  fill(#8B8A8A);  ellipse(750, 2000, 3000, 3000);  //sun  fill(random(0, 255), random(0, 255), random(0, 255));  ellipse(1250, 200, 200, 200);    //sun's rays  fill(random(0, 255), random(0, 255), random(0, 255));  line(1150,40,1190,90);  //craters  fill(0);  ellipse(200, 700, 100, 100);  ellipse(500, 834, 100, 100);  ellipse(800, 750, 100, 100);  ellipse(1200, 750, 100, 100);  ellipse(400, 700, 100, 100);  ellipse(600, 600, 100, 100);  ellipse(1050, 650, 100, 100);  //other planets  fill(#CB8C25);  ellipse(1100, 200, 40, 40);  fill(#EDD0A0);  ellipse(1000, 200, 60, 60);  fill(#232FFA);  ellipse(850,200,80,80);  } |

Notes (Points of interest, mistakes, lessons learned, web resources, and thoughts):

|  |
| --- |
|  |