**APCSP Activity 1.4.1: Procedural Abstraction**

Learning Target: Define the problem and analyze research to create a solution to a problem.

**Step 2**

Define ‘abstraction’ in your own words. Give a real-world example.

Abstraction is basically using something without actually knowing the inner workings of it. An example of this would be a car. You don’t need to know how a car works to know how to use it. The same can be said about computer programming. Someone can write a library of code that other people can use without actually knowing what each line of code in that library actually does with provided decent usage documentation.

**Step 4**

What is high-level documentation? Why is it important?

High-level documentation is an overview of how something works without all of the details. This is important because an API or library of code can’t be used by anyone if nobody knows how it works and high-level documentation would exist to give a brief overview of how the code works and how you can use it to suit your needs.

**Step 5**

What problems are created if people don’t collaborate to create standards?

If nobody collaborated to create standards there would be lots of inconsistent libraries of code which all had vastly different ways to use them. If every library of code worked different than the others then it would be very hard to use lots of these libraries because you would have to relearn different methods of using code for every new library you use. Standards are nice because you can just look at some simple documentation for a library of code and you would easily be able to use it because every other library works the same way.

**Step 6**

The *Python* programming language uses dot notation to refer to the attributes of an object and to call the class’ methods on an object. For example, we used the AxesSubplot class and called set\_title() on ax, which was an AxesSubplot:

ax.set\_title('My picture')

On the right is the AxesSubplot class described with a UML diagram. Add another method from the AxesSubplot that would be listed in this UML class diagram.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *Class name* | AxesSubplot |  | *Class name* | AxesSubplot |
| *Attribute names* | title |  | *Attribute names* |  |
| *Method names* | set\_title() |  | *Method names* | legend() |

**Step 7**

Explain the following line of code:

Screen Shot 2017-11-14 at 3.19.13 PM.png

The “import” keyword is used whenever you want to reference an external library of code. In python, libraries of code can be exported as a specific function or object, and when you import it into a different python file you can define the export function/object as whatever you want. Most of the time this would be used to shorten function names and making the code more readable.

**Step 8**

What is the purpose of fiat?

“Fiat” is the variable name that contains all of what “realityArtist” exported as.

**Step 9**

What is the result of print mike.color?

Mike.color would return “#BB7700” because that was what the color argument was defined as when the PingPongBall object was initiated.

**Step 11**

Write the lines of code would instantiate a ball called marco and then draw black, red, and gray circles on it.

|  |
| --- |
| marco = fiat.PingPongBall() marco.drawCircle(mike, diameter=10, color="#000000", fill=False) marco.drawCircle(mike, diameter=10, color="#FF0000", fill=False) marco.drawCircle(mike, diameter=10, color="#999999", fill=False) |

**Step 12**

Write the piece of code to create a classroom of 6 students. Instantiate more instances of PingPongBall.

|  |
| --- |
| for i in range(6):  items.append(PingPongBall()) |

What is items[1]?

Items[1] is the second value in the items array which was defined as the PingPongBall() constructor function.

What is items[3].color?

items[3].color would return “#BB7700” because that was the default color defined when the object was constructed.

**Step 13**

Describe the code:

|  |
| --- |
| items[3].draw\_circle(myColor = '#00FF00', diameter = 6) |

Using the 4th item in the items array, the code would use the draw\_circle function to draw a circle within that ping pong ball using solid green with a diameter of 6.

**Step 14**

Describe the code:

|  |
| --- |
| for pingpong in items:  pingpong.draw\_circle(diameter=6, fill=True) |

This would run through every ping pong ball in the items array and draw a circle.

**Step 15**

Describe the code:

|  |
| --- |
| for pingpong in items:  pingpong.append(PingPongBall()) |

This would add a ping pong ball to the pingpong array using the PingPongBall() constructor function.

**Step 16**

What are some differences between the GolfBall and the PingPongBall classes?

The GolfBall() constructor function has different default parameters such as the color which is “#FFFFFF” (pure white). They also have similar method names.

**Step 17**

Describe the code:

|  |
| --- |
| for i in range(8):  items.append(GolfBall()) |

This adds 8 golf balls to the items array using the GolfBall() constructor function.

**Step 18**

Describe the code:

|  |
| --- |
| for ball in items:  ball.draw\_circle() |

This would run through every ball in the items array and call the draw\_circle() method in each ball.

**Conclusion Questions**

1. Think of an example from your daily life where you use abstraction. Describe some of the details you discard and some of the generality you gain by using abstraction.

Most of the time people use computers without understanding how they work.

2. What is the difference between procedural abstraction and data abstraction?

Procedural abstraction is when you use something without knowing how it works and data abstraction is just understanding the bare minimum to complete a task(i.e. Object specific variables)

3. The GUI was first developed in 1961 by Ivan Sutherland for his Ph.D. at M.I.T. You might watch a 1964 video produced by MIT, especially the demo of Sutherland’s work starting at 3:20, at <http://www.youtube.com/watch?v=USyoT_Ha_bA>.

Bill Gates at Microsoft got inspiration for Windows from Apple’s Steve Jobs. Steve Jobs at Apple got inspiration for Macintosh from Xerox’s Alan Kay. Alan Kay at Xerox got inspiration for Star from Trygve Reenskaug.

Trygve Reenskaug created the program Autokon with a graphical user interface in 1963 to design ships.

Trygve Reenskaug got inspiration for Autokon from M.I.T.’s Ivan Sutherland.

All along the way, GUI programming, object-oriented programming, and abstraction have been intertwined. Why do you think GUIs, objects, and abstraction have been connected like this in the history of computer science?

When programming object oriented code, everything can be visualized and generally appears less complicated than it really is.