**Computer Science II – Weekly Quiz 03/10/2017**

**Each question below is worth 10 points. Please answer clearly in the space provided.**

**Jake Lorah**

1. Describe a major advantage of using object-oriented programming languages vs their object-based counterparts

An object-oriented computer programming language is centered around the use of objects rather than individual program actions traditionally used to process user input to provide output.

1. Name 3 benefits of using Greenfoot to learn the concepts of Object-Oriented programming.

3 benefits of Greenfoot is its flexible enough for high school and college students, it makes the principles and concepts of object-oriented programming easily accessible through the creation of graphics, and it allows students with no prior programming experience to become quickly engaged.

1. List and describe the 3 major components of the Greenfoot workspace.

The 3 major components of the Greenfoot workspace is The World: This is where objects will be present and program actions will take place during execution. It’s the largest area of the screen and the location of program execution. The Class Diagram: Located on the right side of the workspace. It keeps classes and objects organized and it represents relationships and inheritance. And lastly, the Execution Controls: Buttons located on the bottom (Act, Run, Reset, Speed) and is used for performing actions within the program.

1. Explain the concept of compiling code. What is it used for?

Every time you make an edit to the source code, no matter how small, you must compile the program in order to run the updated code.

1. The icon below is an example of a Greenfoot object (Wombat). Explain its role in Object-Oriented Programming with Greenfoot.

http://www.android100.org/uploadfile/2013/0926/20130926113059989.jpg

This is the Wombat for the Wombat and Leaves Greenfoot program. The Wombat is an object. In Greenfoot, you must use objects. The object is the picture that you drag over to the workspace. That is very essential to the program. Objects include Wombat, Leaves, Rockets, Asteroids, etc…

1. Explain the concept of ***is-a*** relationships. What Greenfoot feature allows you to view this type of a relationship?

The Class diagram shows class-subclass relationships in an is-a fashion. That means each component is connected to its parent class with arrows. The structure of the class diagram is shown in a is-a fashion.

1. Label all of the Greenfoot controls pictured below and describe each.



Act – Each time you click it, the program moves a small amount. The animation only moves one bit. You can keep clicking it to slowly move the animation.

Run – This runs the whole animation. When you press run, you have the option to pause the animation as well.

Reset – This resets everything on the workspace, but not in the class diagram.

Speed – This adjuster changes the speed of the animation.

Compile – This is the compile button. You must always click this button each time you make an edit to the Source Code.

1. In computer programming, what is a class?

Classes are a fundamental component of Object-oriented programming. It acts as a blueprint or a template for creating objects. It defines the data type of the object and how it can behave.

1. Provide 5 examples of Greenfoot classes used in this unit.

5 examples of classes are World, Actor, Wombats, Asteroids, and Rockets.

1. What is Source Code? Name a major benefit when it comes to using Greenfoot as far as Source Code is concerned.

Source Code is all of the syntax, code and commands that make up a program that can be executed by the computer. In a traditional programming environment, the code is entirely written by the developer or programmer. In Greenfoot, most of the source code is generated by the software, however, the code can be manually updated with the built in Java editor.