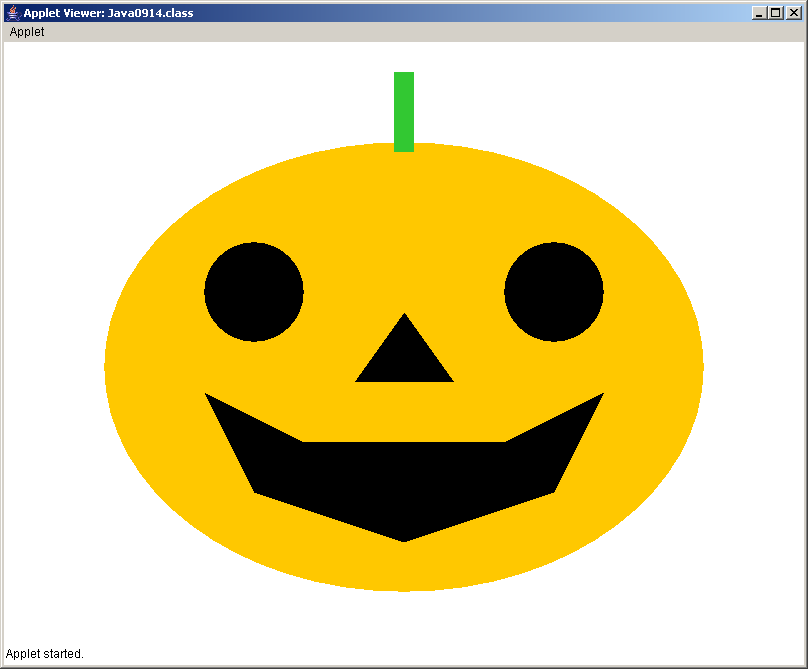
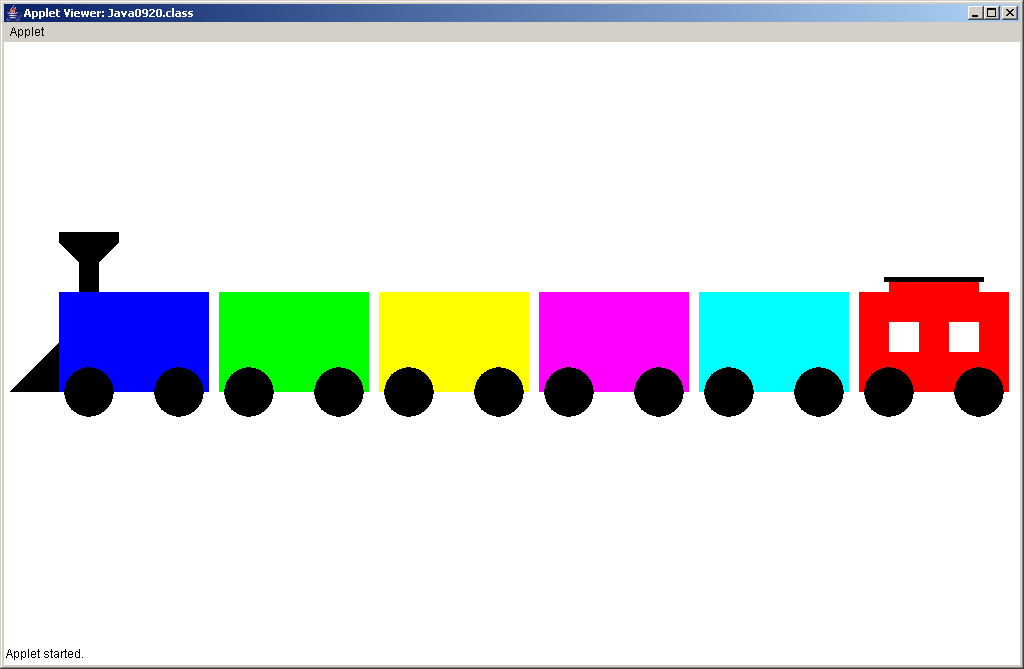
|  |  |
| --- | --- |
| **Exposure Java** | **Lab10** |
| **The Composition/Inheritance**  **Graphics Program** | **80, 90, 100 & 110 Point Versions** |
| **Assignment Purpose:**  This assignment has multiple goals. You need to demonstrate your knowledge of composition and inheritance, demonstrate creativity and demonstrate the ability to work on this assignment with a partner in a small team. | |

Write an **applet** program that demonstrates composition and inheritance. These concepts need to be shown in graphics format similar to the **Tree** examples from chapter 10. Two other examples are shown below: In the case of the **Jack-O’-lantern**, a **Jack-O’-lantern** *is-a* **Pumpkin**, and a **Jack-O’-lantern** *has-a* **Face**. In the case of a **Train**, a **Train** *has-a* **Locomotive**, **TrainCars**, and a **Caboose**; however, a **Locomotive** *is-a* **TrainCar** that *has-a* **Funnel** and a **Scoop**. A **Caboose** *is-a* red **TrainCar** that *has* **Windows** and a **Cupola** (the thing on the top).

This is an **OPEN-ENDED** lab assignment. Frankly, you will find this harder than your previous lab assignments. The required **main or paint** method is not shown. The required program executions for different point versions are not available. This is an assignment written with the creativity of your team or two students. Even though you have control over the classes that you select and the graphics images that you display, there are certain point versions that need to be satisfying certain requirements.

**80 Point Version**

Your program needs to demonstrate one example of inheritance and one example of composition. This will mean a minimum of three separate classes, besides the testing class. You must be aware that the Java syntax has to follow the rules of composition and inheritance. In other words you cannot simply declare some classes that have the required properties in real life. For instance an **Animal** *has-a* **Head** and a **Tiger** *is-an* **Animal**. That is fine, but the Java program statements must correctly use the relationships.

A good example of an 80-point program is the **Jack-O’-lantern** class. This is a simple program that demonstrates inheritance and composition, but does not pass any parameters other than the **Graphics** object **g**.

**90 Point Version**

This program requires at least one example of inheritance and two examples of composition.

**100 Point Version**

This program requires at least two examples of inheritance and three examples of composition.

A good example of an 100-point program is the **Train** class. This is a more advanced program which does require parameter passing for the both inheritance and composition.

**110 Point Version**

Same requirements as the 100 point version, except at least one of the inheritance examples needs to involve *multi-level inheritance* AND pass parameters all the way up to the highest level superclass from the lowest level subclass.