

Unit 1: OO Programming

Task 1: Spinning up to speed...

Welcome back!

It's been a little while since SCC110... so let's take some time to refresh the basics of programming in Java:

- Study the Java Refresher slide deck on the SCC212 moodle page. Make sure you take time to really review the content, and ensure you understand it. I'm considering all the content in there to be a pre-requisite for SCC212.
- Discuss anything you don't fully understand with me or one of the teaching assistants in your lab. If anything comes up that the TAs can't answer (go on... try to think of a question they can't answer!), ask me in the lecture on Thursday. I'll reserve a short slot for such questions.
- Explore the software installed on the lab PCs, and write, compile and execute one or two **very** simple programs of your own invention. Try out some of the new concepts introduced in the refresher slide deck, e.g. enumerations or inline conditionals. I recommend you use Visual Studio Code for SCC212.

When you're comfortable, move on to Task 2 overleaf...

Task 2: Modeling the Solar System

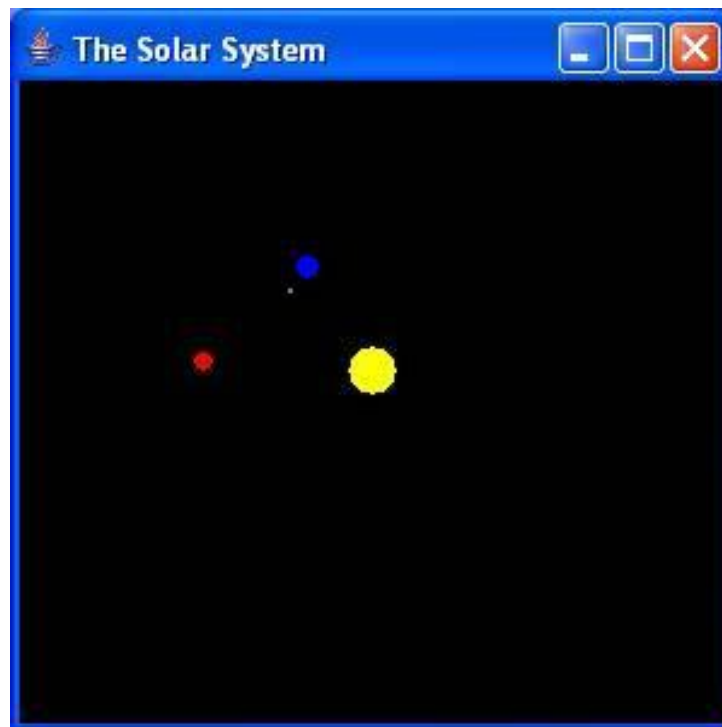
Aim of the Exercise

Over the next few weeks we'll be putting some of the theory about object orientation you learned last year into practice, and then developing it further as we discuss new concepts. You will also be integrating some classes we've written for you with classes of your own design. **The aim is for you to gain experience of developing a well-designed OO program.**

The Task

Your task this week is to create a simple moving model of the solar system, as illustrated below. This isn't a course on GUI programming though, so I've written a class for you that handles the graphical components called **SolarSystem**. You can download the java source file and associated JavaDoc for this class from the associated resources file on Moodle. I've provided the source code for the **SolarSystem** class, but you are **not expected to change the internals of this code**. In fact, for this exercise, **you are not permitted to modify the SolarSystem.java file... just create instances of it.**

Don't forget to take written note of any feedback you gain from teaching assistants on your work, along with your own analysis on what you believe you have learned. List any weaknesses you are aware of in your work, and consider how you plan to address them...



Exploring the SolarSystem.

Based on the SolarSystem class provided and its JavaDoc, write a simple Java program that will:

- Create a window using the SolarSystem class.
- Draw a yellow sun in the centre of the window.
- Draw a blue earth revolving around the Sun.

Hint: You'll need to read and experiment with the **drawSolarObject** and **finishedDrawing** methods.

Analysis:

Consider the limitations of your solution, paying particular attention to *extensibility* and *scalability*...
Write down your thoughts.

Additional Study:

If you've completed the above, add more items to your solar model. For example, all eight planets, and the main moons of those planets.