

## CS224 – Inheritance and Polymorphism

**Purpose:** We will take a close look at inheritance in Java so that we can investigate the mechanisms that support it.

**Knowledge:** This activity will help you become familiar with the following content knowledge:

- The specifics of the inheritance hierarchy in Java

**Activity:** With your group perform the following tasks and answer the questions. You will need to download and import the lab3 code into Eclipse. You will be reporting your answers back to the class in 30 minutes.

1. Carefully examine the classes named A, B, and C. You will see the following:
  - (a) class A defines two fields x and y and two methods sum and product
  - (b) class B extends A overriding field y and defines method power
  - (c) class C extends B overriding product and power and defining field z

Examine the TestCode class. We start by defining a lot of objects with the following naming convention: The first letter indicates what type the object is stored in and the second letter indicates how the object was instantiated. So object bc indicates that we created an object of type C and cast it to type B.

Why would the following cast produce an error?

```
C c0 = (C) new A(2,3);
```

2. When instantiating cc, bb, and aa which constructors were called in which order and why?
3. Which toString() method was called on ac, bc, and cc and why?
4. Which class is printed for aa, ab, ac and why?
5. If ac is an instance of C, and ab is an instance of B then why can't we use the method ac.power()?
6. Observe the different fields for the C and B objects. Notice that there is a B.y field (check it out in the code for B). Why did it "shadow" or override the y variable we put in the constructor? What happens to the results of the power function when we comment out the y field in B and why?
7. Dynamic binding is when the type of the object (and therefore which method to call) is determined when you run the program. Given the results in TestCode, explain how dynamic binding works.