Winston Wang

Mr. Kuszmaul

APCS

11 January 2015

Vocab

1. Inheritance creates a hierarchy of parent and child between two classes, while delegation only allows one class to use the methods of another class.
2. When two classes inherit an abstract function, polymorphism can occur
3. Classes can inherit functions are overloading
4. It is possible to delegate methods that are polymorphic to other classes
5. I can delegate each of a set of overloading methods to different classes
6. Polymorphism and overloading are similar; however, polymorphism involves the same method name across classes, while overloading involves the same method name within the same class.

Inheritance:

**public** **class** Foo

{

**public** Foo()

{

}

}

**public** **class** Foo2 **extends** Foo

{

**public** Foo2()

{

}

}

Foo2 inherits all the methods from Foo

Delegation:

**public** **class** Foo

{

**public** Foo()

{

}

**public** **void** hi()

{

System.***out***.println("hi");

}

}

**public** **class** Foo2

{

**private** Foo delegate;

**public** Foo2()

{

delegate = **new** Foo();

}

**public** **void** hi()

{

delegate.hi();

}

}

The hi() method in Foo2 is delegated to Foo

Polymorphism:

**public** **abstract** **class** Foo

{

**public** Foo()

{

}

**public** **void** hi()

{

}

}

**public** **class** Foo2 **extends** Foo

{

**public** Foo2()

{

}

**public** **void** hi()

{

System.***out***.println("hi 1");

}

}

**public** **class** Foo3 **extends** Foo

{

**public** Foo3()

{

}

**public** **void** hi()

{

System.***out***.println("hi 2");

}

}

The two hi() methods are polymorphic

Overloading:

**public** **abstract** **class** Foo

{

**public** Foo()

{

}

**public** **void** hi()

{

System.***out***.println("hi");

}

**public** **void** hi(**int** i)

{

System.***out***.println("hi"+i);

}

**public** **void** hi(String j)

{

System.***out***.println("hi"+j);

}

}

The three hi() methods are overloading