1A.

**Dog d = new Dog();**

**System.out.println(d.call(3));**

A new dog object is created by using the default constructor. Then the dog object is calling the call function in Dog.java and passing it 3 which prints out woof 3 times.

1B.

**Object c = new Cat();**

**System.out.println(c);**

Overrides the Object toString method and uses Cat’s toString method

1C.

**Object v = new Pet();**

**System.out.println(v.call());**

Returns an error because the object was not instantiated properly by using Pet v = new Pet();

1D.

**Pet p = new Pet();**

**System.out.println(p);**

Prints the memory location of Pet p

1E.

**Pet q = new Dog();**

**System.out.println(q.call());**

Calls the function call() from Dog.java because it was initiated as new Dog(). Overrides Pet’s call function.

1F.

**Cat t = new Cat();**

**System.out.println(t.toString("Come here, kitty"));**

Calls the function toString(string b) from Cat.java.

1G.

**Cat u = new Cat();**

**System.out.println(u.call(3));**

Calls the function toString(string b) from Cat.java but the arguments do not match so it does not work.

1h.

**Dog e = new Dog();**

**System.out.println(e.feet());**

Calls the feet function from parent Pet.java. Example of polymorphism.

1I.

**Pet r = new Dog();**

**System.out.println(r.call(3));**

Calls the function call() from Dog.java, the arguments do not match so there is an error.