

## PATTERNS

1. Which pattern do you use to create instances of `java.inet.InetAddress`?
  - a. Singleton
  - b. Builder
  - c. **Factory**
  - d. Prototype
  - e. Strategy
  
2. 

```
public class Example {  
    private static final Example foo = new Example();  
  
    private Example() {  
    }  
    public static Example getIt() {  
        return foo;  
    }  
  
    // .... Other methods of class  
}
```

Which creational pattern is represented in the sample code above?

- a. Prototype
- b. Builder
- c. Proxy
- d. **Singleton**
- e. Factory

## PATTERNS

3. 

```
public class MyFrame extends JFrame {  
    public MyFrame() {  
        JButton one = new JButton("One");  
        add (one, BorderLayout.NORTH);  
        JButton two = new JButton("Two");  
        add (two, BorderLayout.SOUTH);  
        WindowListener l = new WindowAdapter() {  
            public void windowClosing(WindowEvent e) {  
                System.exit(0);  
            }  
        };  
        addWindowListener(l);  
    }  
}
```

The WindowListener in the sample code above exhibits which behavioral pattern?

- a. Command
- b. Strategy
- c. Chain of Responsibility
- d. Observer**
- e. State

4. Sample code

```
public class Example {  
    private Sample sample;  
  
    public Example() {  
        sample = new Sample(1, 2, 3);  
    }  
}
```

Referring to the sample code above, which action do you take to use a Mock Object for testing?

- a. Pass the arguments of Sample to the Example constructor.
- b. Make the constructor private, and create a factory method.
- c. Make the constructor protected, and only create instances of subclasses.
- d. Pass the created Sample into the constructor.
- e. Change the constructor to use a factory method to create the Sample class.**