

Java Self Test

Each problem is worth 10 points.

1. What is the difference between the “==” operator and the “equals()” method for String objects?
2. In writing a general purpose sort method in Java, the sort method needs to compare objects. There are two ways to give the sort method the code to compare objects. **Briefly**, describe the two ways.

3. Given class A and B below.

```
class A {  
    public void bar() {  
        System.out.println( "A" );  
    }  
}
```

```
class B extends A {  
    public void bar() {  
        System.out.println( "B" );  
    }  
}
```

Parts a and b refer to the following two lines of code:

```
A test = new B();  
test.bar();
```

- a. If all method access in Java were determined statically then what would the two lines of code above print out?
- b. If all method access in Java were determined dynamically then what would the two lines of code above print out?
- c. When is method access in Java determined statically?

4. We have class A and class B, a subclass of A. Class A has a method called foo. Class B overrides the foo method. Briefly, what are the restrictions placed on the return type, the parameters and the exceptions thrown by the foo method in class B?

5. Given the class definition of the class File below, what implications does this have for the file structure, compiling the class, and using the class in a program?

```
package sdsu.whitney.io;
public class File {
    //code not shown
}
```

6. Circle the assignment statements, "Line 1" through "Line 10", in the following code that illegally access a field. Why are they illegal accesses?

```
package Exam;
public class Parent {
    protected String protectedVar = "Protected";
    String packageVar = "Package";
}
```

```
package Exam;
public class Uncle {
    public void SampleAccess( Parent parameter ) {
        parameter.protectedVar = "Line 1";
        parameter.packageVar = "Line 2";
    }
}
```

```
package Quiz;
public class Aunt {
    public void SampleAccess( Exam.Parent parameter ) {
        parameter.protectedVar = "Line 3";
        parameter.packageVar = "Line 4";
    }
}
```

```
package Quiz;
public class Child extends Exam.Parent {
    public void SampleAccess( Exam.Parent parentType,
        Child childType ) {
        parentType.protectedVar = "Line 5";
        parentType.packageVar = "Line 6";

        protectedVar = "Line 7";
        packageVar = "Line 8";

        childType.protectedVar = "Line 9";
        childType.packageVar = "Line 10";
    }
}
```

7. In the following code, what are the values of a, b, c, and name when the System.out.println statement is executed. Explain the output.

```
class DefaultValues {  
    int a;  
    String name;  
  
    public void aMethod() {  
        int b;  
        int[] c = new int[5];  
        System.out.println( "Start Here");  
    }  
}
```

8. What does it mean to say that the ListIterator is fail-fast?

9a. What does the following code print out when we execute "new A()"? Why?

```
class A {
    static int a = 0;

    int b = a++;

    {
        a = a + 2;
    }

    static int c = a;

    static {
        a = a + 5;
    }

    int d = a;

    public A() {
        System.out.println( "a =" + a );
        System.out.println( "b =" + b );
        System.out.println( "c =" + c );
        System.out.println( "d =" + d );
    }

    static {
        a = a + 9;
    }
}
```

9b. Given the classes below, what will be the output of statement "new Child()"?
Why?

```
class Parent {  
    public Parent() {  
        System.out.println( "In Parent");  
    }  
  
    {  
        System.out.println( "In Parent Block");  
    }  
}
```

```
class Child extends Parent {  
  
    {  
        System.out.println( "In Child block");  
    }  
  
    public Child() {  
        super();  
        System.out.println( "In Child");  
    }  
}
```

10a. What will be the result of trying to compiling and executing the following program if `FooException` is properly defined as a subclass of `java.lang.Exception`? Explain.

```
class ExceptionQuestion {
    public void aMethod() {
        System.out.println( "In aMethod" );
        throw new FooException();
    }

    public static void main( String[] args ) {
        try {
            System.out.println( "Start" );
            ExceptionQuestion x = new ExceptionQuestion();
            x.aMethod();
            System.out.println( "After method" );
        } catch ( FooException error ) {
            System.out.println( "In handler" );
        }
        System.out.println( "End" );
    }
}
```


10b. Assume that the exception `FooException` is properly defined. What will be the output of compiling and executing the main method of the following class? Explain.

```
class ExceptionQuestionB {
    public void aMethod() throws FooException {
        try {
            System.out.println( "In aMethod" );
            throw new FooException();
        } catch ( FooException error ) {
            System.out.println( "in first catch" );
            throw new FooException();
        } finally {
            System.out.println( "Finally" );
            return;
        }
    }
}

public static void main( String[] args ) {
    try {
        System.out.println( "Start" );
        ExceptionQuestionB x = new ExceptionQuestionB();
        x.aMethod();
        System.out.println( "After method" );
    } catch ( FooException error ) {
        System.out.println( "In handler" );
    }
    System.out.println( "End" );
}
}
```

11. True or False. If a method is synchronized then all methods that override that method are automatically synchronized.

True or False. If all methods of a class are synchronized, an object of that class cannot be involved in a deadlock.

True or False. A thread cannot change its own priority.

True or False. A thread's type (daemon or user) can not be changed once it has started running.

True or False. Calling a static synchronized method locks all access to any synchronized method on any object of that class.

12. True or False. All threads in Java are time-sliced.

True or False. The `FileReader` and the `FileInputStream` are not buffered.

True or False. A `FileWriter` stores Unicode characters in a file, while a `FileOutputStream` stores ASCII characters in a file.

True or False. Given a try block with a finally, it is not possible to exit the try block without executing the finally block.

True or False. The clone method is required to call `super.clone()`.