Multiple Choice Questions

Answer the following questions on this paper first. Later transfer your answers to a green Scantron sheet. You may keep this first section at the end of the test and compare your answers to my posted answers. You will hand in the other sections.

1. Immediately after this statement executes, what best describes the result:

   ```java
   int[] x;
   ```

   a) \( x \) is a one-dimensional array of \( int \) values.
   b) \( x \) holds location information for a one-dimensional array \( int \) values (default size 10 elements).
   c) \( x \) stores the value null.
   d) \( x \) is a reference variable that now stores the location information for an array of references to \( int \) values.
   e) None of the above.

2. Immediately after this statement executes, what best describes the result:

   ```java
   int[][] x = {{1, 2}, {3, 4, 5}, {6}, {7, 8}};
   ```

   a) \( x \) is a reference variable that now stores the location information for a two-dimensional array of \( int \) values.
   b) \( x \) is a two-dimensional array \( int \) values.
   c) \( x \) is allocated on the stack; the resulting array is also allocated on the stack.
   d) \( x \) is a reference variable that now stores the location information for an array of 4 references to 4 separate arrays of \( int \).
   e) None of the above.

3. Immediately after this statement executes, what best describes the result:

   ```java
   Actor x = new Wizard();
   ```

   a) \( x \) is an object of type \( Wizard \). \( x \) is allocated on the stack.
   b) \( x \) is an object of type \( Wizard \). \( x \) is allocated on the heap.
   c) \( Wizard \) is a subclass of \( x \).
   d) The statement generates an exception.
   e) None of the above.

4. Memory is allocated when an object is created using a new operator. That memory is released when:

   a) The delete operator is called using the reference variable to define the location of the object.
   b) Program execution enters a catch block.
   c) The garbage collector discovers that there are no outstanding references to the object.
   d) The class loader detects an exception.
   e) None of the above.

5. The term signature can be used when describing a method. In this context, a signature is:

   a) The method name and the number of arguments sent to the method.
   b) The method name and the pattern of arguments sent to the method (both the number of and the data type of each argument).
   c) The method name and whether the method is: public, private, or protected.
   d) The method name and the return type for a method.
   e) All of the above.

6. Which of the following is an application of the principle of inheritance:

   a) An object of class \( A \) has a reference to a class \( B \) object.
   b) Several methods have the same name, but have different signatures.
   c) Fields are usually declared private.
   d) Objects created with new are allocated on the heap.
   e) None of the above.

7. A virtual method requires which of the following to be in effect:

   a) The method signatures must be identical in the superclass and subclass.
   b) The subclass overrides all methods in the ultimate superclass (called Object).
   c) The keyword virtual is applied to the method name.
   d) The superclass must be declared to be abstract.
   e) All of the above.

8. Which of the following is true about a primitive variable:

   a) It holds the raw machine-code address of a variable.
   b) During program execution, accessing a primitive is slower than accessing a reference-based object.
   c) Primitives are close to the machine-code level, thus have different byte sizes on different hardware platforms.
   d) The value stored in the variable is an actual value (as opposed to a reference to something).
   e) None of the above.

9. Which of the following is true about a reference variable:

   a) It holds the raw machine-code address of a variable.
   b) When created, it will always contain location information for some object.
   c) It can hold location information for objects of the named class, or objects of any subclass.
   d) Once the location information in the reference variable is established, it cannot be changed.
   e) None of the above.
10. Which of the following statements about constructors is correct:
   a) A constructor has the same name as the class name.
   b) Once execution is inside a constructor, the constructor allocates memory for the instance variables.
   c) Constructor methods have a void return type.
   d) A class can only have one constructor.
   e) All of the above.

11. In the example code fragment shown below, the keyword abstract:

```
public abstract class Test // ... more class code
```

   a) Implies that some other class will be extended (derived) from Test.
   b) Makes class Test independent of all other classes, that is, it is not a subclass of the class Object.
   c) Is needed so that the class Test can implement the Singleton design pattern.
   d) Ensures that only one object of type Test is ever created.
   e) All of the above.

12. Which of the following is true about exception handling:
   a) Error control is managed by returning a boolean value: true for success; false for failure.
   b) When an error occurs, an exception handling object is created by a method and is thrown back to the calling routine.
   c) The exception class must be a subclass of the class where the exception occurs.
   d) As a final step, an exception must be handled with a call to System.exit(0);
   e) None of the above.

13. Which statement will create an array of 100 references to Actor objects:
   a) Actor[100] aaActors = new Actor[100];
   b) Actor aaActors = new Actor[100];
   c) Actor[100] aaActors = new Actor[ ];
   d) Actor[ ] aaActors = new Actor[100];
   e) None of the above.

14. Assume that you do now have a correctly built array of 100 references to Actor objects managed through the variable aaActors. You have also created many different objects of type Hobbit, Wizard, Elf, etc. and stored references to them in the array. Identify the code needed to display the contents of the Hobbit object that is known to reside at index position 2.
   a) System.out.println( ( (Hobbit) aaActors[2]).toString() );
   b) System.out.println( aaActors[2] );
   c) System.out.println( aaActors[2].toString() );
   d) System.out.println( (Hobbit) aaActors[2] );
   e) None of the above.

15. Which of the following statements about serialization is correct:
   a) A class becomes eligible for serialization if it implements the Serializable interface.
   b) Serialization is efficient because it stores or transmits only the data (not the structural information).
   c) The output stream is encrypted and cannot be inspected by anyone probing the stream.
   d) A serialized stream will output the integer-based reference-id values to allow reconstruction during later input operations.
   e) All of the above.
Trace Program Execution

Given the following working Java code show the output that will result:

```java
public class Stuff{
    private static final double x = 0.6;
    private int n;
    private String s;

    public static void main(String[] args) {
        String s = "Test";
        Stuff stuff = new Stuff(s, 11);
        System.out.println(stuff);
        stuff.doSomething(2.5);
        stuff.changeSomething("Play");
        System.out.println(stuff);
    }

    public Stuff(String s, int n) {
        this.s = s;
        this.n = n / 3;
    }

    public void changeSomething(String s) {
        s = s + "ing";
        System.out.println(s);
    }

    public void doSomething(double d) {
        n = (int) d * 2;
        System.out.println(this);
        n = (int) doSomeMore();
    }

    private double doSomeMore() {
        System.out.println(this);
        double d = n * x;
        System.out.println(d);
        return d;
    }

    public String toString() {
        return "Stuff has " + s + n;
    }
}
```

Identify Syntax / Logic Errors

There are 3 syntax and/or logic errors. Circle each error and suggest a correction.

```java
public class Army {
    String sName;
    Actor[] aArmy;

    public Army(String sName, int nNum) {
        sName = sName;
        aArmy = new Actor[nNum];
        for (int n=0; n<=nNum; ++n)
            aArmy[n] = ActorFactory.createActor();
    }

    public String getName() { return sName; }

    public String display() {
        for (Actor a : aArmy)
            System.out.println(a);
    }
}
```
**Draw a Memory Map**

The following code is taken from the lab exercise on serialization. That lab built a set of student records with course information. I show the code below (minus any material about serialization, since that’s not part of this test). I also show input for a sample run of the program.

Draw a memory map that reflects how data is organized. Draw the map on a separate sheet of paper which will be stapled to this exam.

Note: There are no intentional syntax errors in the following code. But in editing for this test, I might have introduced some by accident. If you do discover one, ignore it as an issue for the memory map. But if you provide a correction, you’ll be entitled to a bonus mark.

```java
public class StudentArrayTest {
    public static void main(String[] args) {
        Student[] asClass = new Student[4];
        for (int i = 0; i < asClass.length; ++i) {
            asClass[i] = new Student(Input.getString("First:"), Input.getString("Last:"));
            asClass[i].setGrades();
        }
        for (int i = 0; i < asClass.length; ++i) {
            System.out.println(asClass[i]);
        }
    }
}

public class Student {
    private String sLast;
    private String sFirst;
    private Course[] courseInfo;
    public Student(String sFN, String sLN) {
        sFirst = sFN;
        sLast = sLN;
    }
    public void setGrades() {
        int nNumCourses = Input.getInt("Number of Courses:", 0, 10);
        courseInfo = new Course[nNumCourses];
        for (int i = 0; i < nNumCourses; ++i) {
            courseInfo[i] = new Course();
        }
    }
}

public class Course {
    private String sCourseName;
    private int nGrade;
    public Course() {
        sCourseName = Input.getString("Course Name:");
        nGrade = Input.getInt("Grade: ", 0, 100);
    }
    public String toString() {
        return sCourseName + ":" + nGrade;
    }
}

Sample Data to be Input

First:Bob
Last:Billings
Number of Courses: (0-10): 1
Course Name:Hava Java
Grade: (0-100): 100

First:Betty
Last:Billings
Number of Courses: (0-10): 2
Course Name:COBOL
Grade: (0-100): 50
Course Name:Hava Java
Grade: (0-100): 100

First:Boris
Last:Botski
Number of Courses: (0-10): 3
Course Name:COBOL
Grade: (0-100): 66
Course Name:Hava Java
Grade: (0-100): 67
Course Name:Networking
Grade: (0-100): 93

First:Bill
Last:Best
Number of Courses: (0-10): 1
Course Name:English
Grade: (0-100): 72

public class String toString() {
    String sInfo = sLast + ", " + sFirst + ": ";
    for (int i = 0; i < courseInfo.length; ++i) {
        sInfo += courseInfo[i] + " ";
    }
    return sInfo;
}

} // end class Student
```

CST8284: Object-Oriented Programming I

Midterm: Part I: Winter 2010