CS211 Summer 2012

Dr. Kinga Dobolyi

Exam 1

Student Name: _____

Student G#: _____

Student signature for Honor Code:

_____

Part 1: Short Answer.

1. Draw what memory looks like after each of the following statement blocks (sub-parts
   are independent). Show garbage collection by crossing out items (i.e. do no erase
   anything for this question): (8 points)

   a.   `int x = 3;`
          `String s = "hello";`

   b.   `Person p;`

          `p = new Person(14,22204);`

          `//person stores an int age and int zipcode`

   c.   `Person p;`

          `Person q = new Person(14,22204);`

          `//person stores an int age and int zipcode`

2. What are the results of the following **expressions**? Give the Java type and the value of the expression. If there are multiple expressions/statements in the question, give the value of the last expression. You can make up any reasonable value for the result of a `toString` method call.
(14 pts)

     a. `12 / 5.0`   _____   _____

     b. `'5' + "3"`   _____   _____

     c. `5 % 3`   _____   _____

     d. `String x = "hello";`

       `x.toString();`   _____   _____

     e. `5 + 3 / 1`   _____   _____

     f. `false || true`   _____   _____

     g. `"7" + 3`   _____   _____

3. Give the output of the following code (20 points):

```
public class Test{

    public static void main(String[] args){
        int x = 7;
        float y = 8;
        Person p1 = new Person("Sally", 23);
        Person p2 = new Person("Joe", 11);
        Person p3 = p1;

        System.out.println(y);
        System.out.println(p1);
        System.out.println(p2);
        System.out.println(p3);

        p1 = changeMe(p1,x,y);

        System.out.println(x);
        System.out.println(p1);
        System.out.println(p2);
        System.out.println(p3);

        p1 = p1;
        p3 = p1;
        System.out.println(p2);
        System.out.println(p3);
    }

    public static Person changeMe(Person p, int x,
              float q){
        x--;
        p.setName("Bob");
        p = new Person("Kim", x);
        return p;
    }
}
```

(answer next page)

4. You are given the compiled version of the **Exam1** class below, in some folder on your hard drive: (6 pts)

```java
public class Exam1{
        public final String name = "Kinga";
        private int age;

        public Exam1(int a){
                age = a;
        }

        public String foo(){
                format();
                return name;
        }

        private void format(){
                age = 12;
        }

}
```

Now imagine you have the following code, in a file called **MyClass.java**, that exists in the same folder as your **Exam1.class** file. You try to compile the file below, but it doesn't work. In the table below, list all the line numbers that would generate compilation errors, and then in one sentence or less, explain why that would be a compilation error.

```
1   public MyClass{
2       private String s;
3       public static void main(String[] args){
4           Exam1 ex1 = new Exam1();
5           Exam1 ex2 = new Exam1(3);
6           ex1 = "Hello";
7
8           ex1.foo();
9           ex1.name = "David";
10          s = "Joe";
11          ex3.format();
12
13          System.out.println(ex1.name);
14          System.out.println(ex1.age);
15      }
16
17 }
```

Line number	Reason for compilation error

5. A. Write a public class called **Car** according to the following specifications:
   a. The class should have two mutable attributes, a **String** *model* and a primitive integer *mileage*.
   b. The class should have a default constructor that sets the *mileage* to 0 and the *model* to **NONE**
   c. The class should have another constructor that initializes both fields with arguments passed in through the constructor
   d. The class should have a **toString** method that returns a nicely formatted (your choosing) representation of the contents of a **Car** object as a **String** object.
   e. The class should have a method to increase the mileage by one mile. This method takes no arguments.
   f. The class should always ensure that the *mileage* is a positive integer less than 100000.
   g. The class should use good object-oriented design (22 pts)

(use this space for 6 if you need it)

Part 2: True/False (circle one), and for full credit **_justify_** all of your choices (7 pts)

1. A method's arguments are garbage collected after the method completes

   1.TRUE   FALSE

2. The `short` type is the smallest integer type we can have.

   2.TRUE   FALSE

3. The Java compiler would flag an error where you tried to divide the integer 3 by zero.

   3.TRUE   FALSE

4. The java.lang package is automatically imported for you in Java programs that you write

   4.TRUE   FALSE

5. A static method can reference class instance data

   5.TRUE   FALSE

6. The toString() method can still be called on any object of any type, even if one is not written for that specific type.

   6.TRUE   FALSE

7. I always write my name on exams   7.TRUE   FALSE