Java Midterm - Summer 2012

Question 1:

Complete statements (a) and (b), and answer questions (c) and (d):

(a) An object is an instance of a

(b) The collection of an object's instance variables is known as its

(c) In Java, objects are created with what operator?

(d) In Java, a class may have instance variables and methods. It may also have named blocks of code, at most one of which is executed when an object is created. What are these named blocks of code called?

Answer

(a) **class**

(b) state

(c) new

(d) constructors

Question 2:

A TangoDancer has a name and may have a partner who is also a TangoDancer. If TangoDancer A chooses TangoDancer B to be a partner, the operation succeeds if neither A nor B already has a partner, and A is not B. On the next page, write the choosePartner method and additional methods as desired so that the following DrJava interactions work:

```
> TangoDancer rita = new TangoDancer("senorita rita");
> TangoDancer tony = new TangoDancer("mr tony");
> rita.getName()
"senorita rita"
> rita.hasAPartner()
false
> tony.hasAPartner()
false
> rita.choosePartner(tony)
true
> rita.hasAPartner()
true
> rita.getPartner().getName()
"mr tony"
> tony.hasAPartner()
true
> tony.getPartner().getName()
"senorita rita"
> tony.choosePartner(rita)
false
> TangoDancer elana = new TangoDancer("ms elana");
> elana.choosePartner(elana)
false
```

Answer

```
public class TangoDancer {
      private String name;
      private TangoDancer partner;
      public TangoDancer(String name){
```

```
            this.name = name;
            partner = null;
      }
      public String getName() { return name; }
      public TangoDancer getPartner() { return partner; }
      public boolean hasAPartner() { return partner != null; }
      public boolean choosePartner(TangoDancer other){
            if (other == this)
                  return false;
            if (partner != null)
                  return false;
            if (other.hasAPartner())
                  return false;
            partner = other;
                  other.setPartner(this);
            return true;
      }
      void setPartner(TangoDancer chooser){
            partner = chooser;
      }
}
```

Question 3

For each code segment below, determine how many times the body of the loop is executed. Write one of the following answers after each: 0, 1, infinite, or > 1. Note that "> 1" means more than once but not infinite.

(a)
```
for(int x=1; x<10; x++){
            System.out.println(x);
      }
```

(b)
```
int x=1;
      while(x<10){
            System.out.println(x);
      }
```

(c)
```
int x=1;
      do{
            x = x*2;
      } while(x>=8);
```

(d)
```
int x=10;
      while(x<10){
            System.out.println(x);
            x=x-1;
      }
```

(e)
```
int x=1;
      while(x!=10){
            x = x*2;
      }
```

Answer

(a) >1

(b) infinite
(c) 1
(d) 0
(e) infinite

Question 4 -

Complete the method sum that takes two arrays of integers as arguments. It returns null if either argument is null or if the arrays are not the same length. Otherwise it returns a new array, each of whose elements equals the sum of the corresponding elements in the input arrays. The input arrays should not be changed by the method.

```
class Summation {
	public static int[] sum(int[] one, int[] two)
	{
		if (one == null || two == null)
			return null;
		if (one.length != two.length)
			return null;
		int[] result = new int[one.length];
		for (int i = 0; i < one.length; i++)
			result[i] = one[i] + two[i];
		return result;
	}
}
```

Question 5:

Assuming that the following classes have been defined:

```
	public class A
	{
		public static void method1()
		{
			System.out.println("C1");
		}
	}

	public class B extends A
	{
		public static void method2()
		{
			System.out.println("A2");
		}
	}
	public class C extends B
	{
		public static void method1()
		{
			System.out.println("B3");
		}
	}
```

In the table below, indicate in the right-hand column the output produced by the statement in the left-hand column. If the statement causes an error, fill in the right-hand column with either the phrase "compiler error" or "runtime error" to indicate when the error would

be detected.

Statement	Output
A a = new B();	-----No output----
B b = new B();	-----No Output----
B c = new C();	-----No Output----
a.method1();	-----C1----------
b.method21();	----Error---------
b.method1();	-----C1----------
c.method2();	-----A2----------
c.method11();	----Error---------
a.method2();	----Error---------

Question 6:

For each of the following program fragments, what is the value of the variable x after the statements execute? Treat each expressions as separate code block

```
(a)   int y = 10;
      int x = y;
      y = x * 3;
(b)   int x = 10;
      x = x / 2;
      x = y + 3 * x - 3;
(c) double x = 7 / 2;
(d) boolean x = ( 2 < 3 ) || (5 < 4);
```

(a) Answer: 10
(b) Answer: 17
(c) Answer: 3.0
(d) Answer: true

Question 7:

Write a class that accepts an ordinary number and outputs its equivalent Roman numeral. The ordinary numbers and their equivalent Roman Numerals are given below:

Ordinary Numbers	Roman Numerals
1	I
5	V
10	X
50	L
100	C
500	D
1000	M

Your class should declare properties and methods to handle conversion of an ordinary number to roman numerals. It does not have to handle input and output methods.

```
class Decimal {
       int decimalNo [] = {1000, 500, 100, 50, 10, 5, 1};
       String romanNo[] = {"M", "D", "C", "L", "X", "V", "I"};
       int decimalInp;

       Decimal(int decimalInp) {
              this.decimalInp = decimalInp;
```

```
        }

        public String convertDecimal() {
                String roman = "";
                for (int i = 0; i < romanNo.length; i++) {
                        while (this.decimalInp >= decimalNo[i]) {
                        this.decimalInp -= decimalNo[i];
                        roman  += romanNo[i];
                                }

                }
           System.out.println("Decimal Input: " + this.decimalInp);
                System.out.println("Roman Output: " + roman);

                return roman;
        }
}

class Driver {
        public static void main (String[] args) {
                Decimal decimal = new Decimal(1456);
                decimal.convertDecimal();
        }
}
```

Question 8:

Write a function to compute the square root of a number to a precision of 4.

```
public class NewtonsSquareRoot{
  private static final double EPSILON = .00001;
  private int myNumber;
  private double root;
  private double guess;

  public NewtonsSquareRoot(int number){
       myNumber = number;
  }

  public int getNumber(){
       return myNumber;
  }

  public double findSquareRoot(){
       guess = 1;
       root = Math.sqrt(myNumber);

       while (EPSILON < Math.abs(Math.pow(root, 2) - myNumber))
       {
         guess++;
       }
       return root;
  }

  public void setNumber(int number){
       myNumber = number;
  }

  public String toString(){
```

```
      String s = new String();
      s = "The square root of " + myNumber + " is " + root + ".";
   return s;
  }
}
```