STRING, UTIL CLASSES & INTERNATIONALIZATION

1. Float f=new Float(3.1);
   Integer i=new Integer(1);
   long m=2;

   System.out.println("Result is "+m+f+i);

   What is the output of the sample code above?
   a. Result is 4.12
   b. Result is 5.11
   c. Result is 6.1
   d. Result is 23.11
   e. Result is 24.11

2. String str1= "My name is Billy."
   String str2= "My name is Billy."

   Referring to the sample code above, which code string do you use to compare the strings in an "if" statement, checking only to see if the strings have same characters?
   a. if ((str1 – str2) ==0){
   b. if (str1.compareTo(str2)){
   c. if (str1 ==str2) {
   d. if (str1 <> str2) {
   e. if (str1.equals(str2)) {

3. Which code fragment returns a string representation of j, where j is an int?
   a. ((Object j).newString(j)
   b. Str(j)
   c. new String(j)
   d. (String) j
   e. new integer(j).toString()

4. String currencyValue="10.000,25 DM"; // German format for currency
   NumberFormat nf =
   NumberFormat.getCurrencyInstance(Locale.GERMANY);
   Based on the sample code above, how do you return the value of the string as a double?
   a. Double.parseString(currencyValue,nf);
   b. sscanf(currencyValue, "%00.000,00d",nf);
   c. Locale.GERMANY.parseCurrency(currencyValue).toDouble();
   d. nf.getDouble(currencyValue).(Locale.GERMANY);
   e. nf.parse(currencyValue).doubleValue();

5. Which code fragment do you use to generate a random whole number from 1 to 100?

```
a. int i = Random.nextInt(100) + 1;
b. Random r = new Random();
   int i = r.nextInt(100) + 0.5;
c. Random r = new Random();
   int i = r.nextInt(100) + 1;
d. int i = Math.random(100) + 0.5;
e. int i = (int)(Math.random() * 100);
```

6. Sample Code

```
class Class1 {
    static void fix(String s) {
          String t = s;
          t = t.trim();
          t = t.replace(' ', '_');
          s = t;
    }
    public static void main(String args[]) {
         String x = "> This is a test <";
         fix(x);
         System.out.println(x);
     }
}
```

What is the output of the sample code above?

```
a. > This is a test <
b. >Thisisatest<
c. >This_is_a_test<
d. >_This_is_a_test_<
e. >_This is a test <
```

7. Sample code

```
public class Test {
    public static void main(String[] args) {
         StringBuffer[] messages = new StringBuffer[5];
         messages[0].append("Hello, World!");
         System.out.println("First message is " +
         messages[0]);
    }
}
```

What is the result of the sample code above?

a. First message is null.
b. A NullPointerException is thrown.
c. The code does not compile.
d. An ArrayIndexOutOfBounds is thrown.
e. First message is Hello, World!

8. Code

```
String test = "";
 for(int i = 0; i < 50000; i++) {
      test += "abc";
}
```

How do you rewrite the sample code above to optimize linear concatenations?

a.
```
StringBuffer test = new StringBuffer();
for(int i = 0; i < 50000; i++) {
      test.append("abc");
}
```

b.
```
StringBuffer test = new StringBuilder();
for(int i = 0; i < 50000; i++) {
      test.append("abc");
}
```

c.
```
StringBuilder test = new StringBuilder();
for(int i = 0; i < 50000; i++) {
      test.append("abc");
         }
```

d.
```
String test = new String();
for(int i = 0; i < 50000; i++) {
      test.append("abc");
}
```

e.
```
String test = new String("");
for(int i = 0; i < 50000; i++) {
      test += "abc";
}
```

9. You perform localization by:

a. loading all the libraries that were not previously loaded.
b. translating the code into the local language.
c. removing all components other than the local components.
d. adding locale-specific components and input methods.
e. changing the time stamp of the local machine.

10.
```
public void printIt(String txt) {
      Pattern wordBreakPattern = Pattern.compile("[\\s]");
      String words[] = wordBreakPattern.split(txt);
      for (String word: words) {
           System.out.println(word);
      }
}
```
Referring to the sample code above, what is the result when you invoke the following statement?
```
printIt("Hello\nWorld\t!" );
```
   a. prints:
      Hello
      World!
   b. Throws java.util.NoSuchElementException.
   c. prints:
      Hello
      World
      !
   d. prints:
      Hello
      World
   e. prints:
      Hello
      World
      !

11. Sample Code
```
RoundingMode mode = RoundingMode.????;
BigDecimal big1 = new BigDecimal(-11);
BigDecimal big2 = new BigDecimal(2);
System.out.println(big1.divide(big2, mode));
```
Based on the sample code above, which enumerated type of RoundingMode do you insert in place of ???? to produce a rounding behavior resulting in -5?
   a. UNNECESSARY
   b. CEILING
   c. HALF_EVEN
   d. UP
   e. HALF_UP

12.
```
class Person implements Comparable<Person> {
      private static final Collator collator =
 Collator.getInstance(Locale.ITALY);
      private final String lastname;
      private final CollationKey key;
      Person(String lastname) {
           this.lastname = lastname;
      }
      public int compareTo(Person person) {
            return key.compareTo(person.key);
      } }
```

How do you get the sample code above to execute?

a. Remove "Collator.getInstance(Locale.ITALY);" from the class.
b. Remove "private final CollationKey key;" from the class.
c. Add "int compareTo=Person(String lastname)" to the Person constructor.
d. Add "Collate.getcollator = this.(lastname);" to the Person constructor.
e. Add "this.key = collator.getCollationKey(lastname);" to the Person constructor.