MULTIPLE-CHOICE QUESTIONS ON THE LAB CONCEPTS

1. For ticket-selling purposes, there are three categories at a certain theater.

Age	Category
65 or above	Senior
From 18 to 64 inclusive	Adult
Below 18	Child

Which of the following code segments will assign the correct string to `category` for a given integer age?

```
  I if (age >= 65)
          category = "Senior";
    if (age >= 18)
          category = "Adult";
    else
          category = "Child";

 II if (age >= 65)
          category = "Senior";
    if (18 <= age <= 64)
          category = "Adult";
    else
          category = "Child";

III if (age >= 65)
          category = "Senior";
    else if (age >= 18)
          category = "Adult";
    else
          category = "Child";
```

(A) I only
(B) II only
(C) III only
(D) II and III only
(E) I, II, and III

2. What is the output of the following code segment?

```
String s = "How do you do?";
int index = s.indexOf("o");
while (index >= 0)
{
    System.out.print(index + " ");
    s = s.substring(index + 1);
    index = s.indexOf("o");
}
```

(A) `1 3 2 3`
(B) `2 4 3 4`
(C) `1 5 8 12`
(D) `1 5 8 11`
(E) No output because of an `IndexOutOfBoundsException`

3. Consider the following method `removeAll` that creates and returns a string that has stripped its input phrase of all occurrences of its single-character `String` parameter `ch`.

```
Line 1:  public static String removeAll(String phrase, String ch)
Line 2:  {
Line 3:      String str = "";
Line 4:      String newPhrase = phrase;
Line 5:      int pos = phrase.indexOf(ch);
Line 6:      if (pos == -1)
Line 7:          return phrase;
Line 8:      else
Line 9:      {
Line 10:         while (pos >= 0)
Line 11:         {
Line 12:             str = str + newPhrase.substring(0, pos - 1);
Line 13:             newPhrase = newPhrase.substring(pos + 1);
Line 14:             pos = newPhrase.indexOf(ch);
Line 15:             if (pos == -1)
Line 16:                 str = str + newPhrase;
Line 17:         }
Line 18:         return str;
Line 19:     }
Line 20: }
```

The method doesn't work as intended. Which of the following changes to the `removeAll` method will make it work as specified?

(A) Change Line 10 to
`while (pos >= -1)`

(B) Change Line 12 to
`str = str + newPhrase.substring(0, pos);`

(C) Change Line 13 to
`newPhrase = newPhrase.substring(pos);`

(D) Change Line 14 to
`pos = phrase.indexOf(ch);`

(E) Change Line 16 to
`str = str + newPhrase.substring(pos + 1);`

4. A programmer has written a program that "chats" to a human user based on statements that the human inputs. The program contains a method `findKeyWord` that searches an input statement for a given keyword. The `findKeyWord` method contains the following line of code.

```
pos = statement.indexOf(word);
```

Suppose `pos` has a value `>= 0`, that is, `word` was found. The programmer now wants to test that an actual word was found, not part of another word. For example, if "cat" is the keyword, the programmer needs to check that it's not part of "catch" or "category." Here is the code that tests if `word` is a stand-alone word. (You may assume that `statement` is all lowercase and contains only letters and blanks.)

```
pos = statement.indexOf(word);
//Check for first or last word
if (pos == 0 || pos + word.length() == statement.length())
{
    before = " ";
    after = " ";
}
else
{
    before = statement.substring(pos - 1, pos);
    after = statement.substring(pos + word.length(),
            pos + word.length() + 1);
    if (/* test */)
        //then a stand-alone word was found ...
    else
        //word was part of a larger word
}
```

Which replacement for `/* test */` will give the desired result?

(A) `(before < "a" || before > "z") && (after < "a" || after > "z")`

(B) `(before > "a" || before < "z") && (after > "a" || after < "z")`

(C) `(before.compareTo("a") < 0 && before.compareTo("z") > 0) || (after.compareTo("a") > 0 && after.compareTo("z") < 0)`

(D) `(before.compareTo("a") > 0 && before.compareTo("z") < 0) && (after.compareTo("a") > 0 && after.compareTo("z") < 0)`

(E) `(before.compareTo("a") < 0 || before.compareTo("z") > 0) && (after.compareTo("a") < 0 || after.compareTo("z") > 0)`

5. A program that simulates a conversation between a computer and a human user generates a random response to a user's comment. All possible responses that the computer can generate are stored in an array of `String` called `allResponses`. The method given below, `getResponse`, returns a random response string from the array.

```
/** Precondition:  array allResponses is initialized with strings.
 *  Postcondition: returns a random response from allResponses.
 */
public String getResponse();
{ /* implementation */ }
```

Which is a correct `/* implementation */`?

(A)
```
int i = (int) (Math.random() * allResponses.length);
return allResponses[i];
```
(B)
```
return (String) (Math.random() * allResponses.length);
```
(C)
```
int i = Math.random() * allResponses.length;
return allResponses[i];
```
(D)
```
int i = (int) (Math.random() * (allResponses.length - 1));
return allResponses[i];
```
(E)
```
return (int) (Math.random() * allResponses.length);
```

Questions 6 and 7 refer to the `Deck` class described below.

A `Deck` class contains an array `cards` with an even number of `Card` values and a `final` variable `NUMCARDS`, which is an odd integer.

6. Here are two possible algorithms for shuffling the deck.

Algorithm 1

```
Initialize an array of Card called shuffled of length NUMCARDS.
Set k to 0.
For j=0 to NUMCARDS/2-1
    - Copy cards[j] to shuffled[k]
    - Set k to k+2
Set k to 1.
For j=NUMCARDS/2 to NUMCARDS-1
    - Copy cards[j] to shuffled[k]
    - Set k to k+2
```

Algorithm 2

```
Initialize an array of Card called shuffled containing NUMCARDS slots.
For k=0 to NUMCARDS-1
    - Repeatedly generate a random integer j from 0 to NUMCARDS-1,
        until cards[j] contains a card not marked as empty
    - Copy cards[j] to shuffled[k]
    - Set cards[j] to empty
```

Which is a false statement concerning Algorithms 1 and 2?

(A) A disadvantage of Algorithm 1 is that it won't generate all possible deck permutations.

(B) For Algorithm 2, to determine the last element shuffled requires an average of `NUMCARDS` calls to the random number generator.

(C) Algorithm 2 will lead to more permutations of the deck than Algorithm 1.

(D) In terms of run time, Algorithm 2 is more efficient than Algorithm 1.

(E) If Algorithm 1 is repeated several times, it may return the deck to its original state.

7. The following `shuffle` method is used to shuffle the cards in the `Deck` class.

```
Line 1:  public void shuffle()
Line 2:  {
Line 3:      for (int k = NUMCARDS; k > 0; k--)
Line 4:      {
Line 5:          int randPos = (int) (Math.random() * (k + 1));
Line 6:          //swap randomly selected card with card at position k
Line 7:          Card temp = cards[k];
Line 8:          cards[k] = cards[randPos];
Line 9:          cards[randPos] = temp;
Line 10:     }
Line 11: }
```

The method does not work as intended. Which of the following changes should be made to correct the method?

(A) Replace Line 3 with

`for (int k = NUMCARDS; k >= 0; k--)`

(B) Replace Line 3 with

`for (int k = NUMCARDS - 1; k > 0; k--)`

(C) Replace Line 3 with

`for (int k = 1; k <= NUMCARDS; k++)`

(D) Replace Line 5 with

`int randPos = (int) (Math.random() * k);`

(E) Replace Lines 7 – 9 with

```
Card temp = cards[randPos];
cards[randPos] = cards[k];
cards[k] = temp;
```

Questions 8 and 9 refer to the following.

A word creation game uses letter tiles, where each tile has a letter and a point value for scoring purposes. A `Tile` class is used to represent a letter tile.

```
public class Tile
{
    private String letter;
    private int pointValue;

    //Constructors and other methods are not shown.
}
```

8. The `Tile` class contains a `toString` method that creates a `String` containing the letter and point value of a `Tile`. The string should be in the following format.

```
Letter letter (point value = pointValue)
```

For example,

```
Letter A (point value = 1)
Letter Z (point value = 10)
```

Consider the `toString` method below.

```
public String toString()
{
    return /* code */
}
```

Which `/* code */` leads to correct output?

(A) `Letter + "letter " + "(point value = " + pointValue + ")";`
(B) `"Letter " + letter + ("point value = " + pointValue);`
(C) `Letter + this.letter + " (point value = " + pointValue + ")";`
(D) `"Letter " + letter + " (point value = " + (String) pointValue + ")";`
(E) `"Letter " + letter + " (point value = " + pointValue + ")";`

9. Any two tiles in the word game that have the same letter also have the same point value, but the opposite is not necessarily true. For example, all the vowels have a point value of 1. Two tiles are said to match if they have the same letter. Consider the following `matches` method for the `Tile` class.

```
/** Returns true if the letter on this tile equals the letter
 *  on otherTile. */
public boolean matches(Tile otherTile)
{ return /* code */; }
```

Which replacements for `/* code */` return the desired result? Note: You may not assume that the `Tile` class has its own `equals` method.

I `letter == otherTile.letter`

II `this.equals(otherTile)`

III `letter.equals(otherTile.letter)`

(A) I only
(B) II only
(C) III only
(D) II and III only
(E) I and III only

10. Consider the following method.

```
public static void alterArray(int[] arr)
{
    int mid = arr.length/2;
    for (int i = 0; i < mid; i++)
    {
        int temp = arr[i];
        arr[i] = arr[arr.length - i - 1];
        arr[arr.length - i - 1] = temp;
    }
}
```

If the current state of a matrix `mat` is

```
2 7 9 5
8 1 4 3
6 5 0 9
```

which matrix will result from the method call `alterArray(mat[2])`?

(A)
```
2 7 9 5
3 4 1 8
6 5 0 9
```

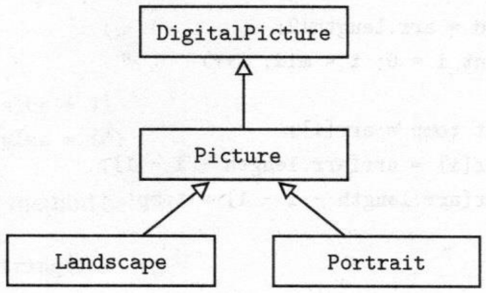
(B)
```
2 7 0 5
8 1 4 3
6 5 9 9
```

(C)
```
5 9 7 2
3 4 1 8
9 0 5 6
```

(D)
```
2 7 9 5
8 1 4 3
9 0 5 6
```

(E)
```
5 9 7 2
8 1 4 3
6 5 0 9
```

11. Consider a program to manipulate digital images. The inheritance hierarchy is as follows.



You may assume that `DigitalPicture` and `Picture` have default (no-argument) constructors, but that `Landscape` and `Portrait` do not have any constructors. Which of the following declarations will compile?

```
  I DigitalPicture p = new Portrait();
 II Landscape p = new Picture();
III DigitalPicture p = new DigitalPicture();
```

(A) I only
(B) II only
(C) III only
(D) I and II only
(E) I and III only

12. A `Pixel` class has several mutator methods that allow the color of a `Pixel` to be changed. For example,

```
/* Sets amount of red in Pixel to value. */
public void setRed(int value)
{ /* implementation not shown */ }
```

Consider a `Picture` class that has a private instance variable `pixels`, which is a 2D array of `Pixel` objects. There are also `int` variables `rows` and `cols` that contain the number of rows and columns in the `pixels` array.

A method `removeRed` in the `Picture` class sets the red value of every pixel to zero.

```
public void removeRed()
{
    for (int row = 0; row < numRows; row++)
        for (int col = 0; col < numCols; col++)
        {
            /* code to set red value to 0 */
        }
}
```

Which is a correct replacement for `/*` ***code to set red value to 0*** `*/`?

```
I   Pixel p = pixels[row][col];
    p.setRed(0);

II  pixels[row][col].setRed(0);

III pixels[row][col] = 0;
```

(A) I only
(B) II only
(C) III only
(D) I and II only
(E) I, II, and III

13. Consider a class `MatrixStuff` that has a private instance variable `mat`.

```
private int[][] mat;
```

The following method uses a vertical mirror down the center of a matrix to reflect the left half of the matrix onto the right. The following two examples show the result of mirroring a two-dimensional array of numbers from left to right vertically. (Another way of saying this is that the right half of the matrix is replaced by a vertical mirror image of the left half.)

Example 1:

```
                mat                          mat after mirroring

 1     2     3     4     5           1     2     3     2     1
 6     7     8     9    10           6     7     8     7     6
11    12    13    14    15          11    12    13    12    11
```

Example 2:

```
              mat                      mat after mirroring

 1     2     3     4               1     2     2     1
 5     6     7     8               5     6     6     5
 9    10    11    12               9    10    10     9
```

```
public static void mirrorVerticalLeftToRight(int[][] mat)
{
    int width = mat[0].length;
    int numRows = mat.length;
    for (int row = 0; row < numRows; row++)
        for (int col = 0; col < width/2; col++)
            /* element assignments */
}
```

Which replacement for `/*` ***element assignments*** `*/` will make the method work as intended?

(A) `mat[row][col] = mat[row][width - col];`

(B) `mat[row][width - col] = mat[row][col];`

(C) `mat[row][width - 1 - col] = mat[row][col];`

(D) `mat[row][col] = mat[row][width - 1 - col];`

(E) `mat[row][width - 1 - col] = mat[col][row];`

14. Consider a square matrix in a class that has a private instance variable `mat`.

```
private int[][] mat;
```

Method `alter` in the class changes `mat`.

```
public void alter()
{
    for (int row = 1; row < mat.length; row++)
        for (int col = 0; col < row; col++)
            mat[col][row] = mat[row][col];
}
```

If `mat` has current value

```
{{1, 2, 3},
 {4, 5, 6},
 {7, 8, 9}}
```

what are the contents of `mat` after method `alter` has been executed?

```
(A) {{1, 4, 7},
     {4, 5, 8},
     {7, 8, 9}}

(B) {{1, 4, 7},
     {2, 5, 8},
     {3, 6, 9}}

(C) {{1, 2, 3},
     {2, 5, 6},
     {3, 6, 9}}

(D) {{9, 6, 3},
     {8, 5, 6},
     {7, 8, 9}}

(E) {{1, 2, 3},
     {4, 5, 2},
     {7, 4, 1}}
```