Sample Multiple-Choice Question 1

01. Consider the following method.

```java
/** Precondition: list contains [23, 39, 42, 28, 99, 57]
 */
public static void mystery(ArrayList<Integer> list)
{
    for (int k = 0; k < list.size(); k++)
    {
        if (list.get(k) % 2 != 0)
            list.remove(k);
    }
}
```

Which of the following describes the values stored in list after mystery executes?

(A) [39, 42, 28, 57]
(B) [42, 28]
(C) [23, 39, 99, 57]
(D) [23, 39, 42, 28, 99, 57]
(E) An IndexOutOfBoundsException error message
Sample Multiple-Choice Question 2

02. Consider the following code segment.

```java
int n1 = Integer.MAX_VALUE;
int n2 = Integer.MIN_VALUE;
int result = n1 + n2;
System.out.println(result);
```

What will be printed as a result of printing the segment?

(A) 1
(B) 0
(C) -1
(D) compile error indicating that the integer number is too large
(E) An ArithmeticException error message
Sample Multiple-Choice Question 3

03. Consider the following code segment.

```
int[][] matrix = {{11,22,33},{44,55,66},{77,88,99}};
for (int k = 0; k < 3; k++)
{
    matrix[k][k] = matrix[0][0];
}
```

Which of the following describes the contents of `matrix` after the code segment executes?

(A) All of the integers in the left-bottom to top-right diagonal are the same.
(B) All of the integers in the top row are the same.
(C) All of the integers in the bottom row are the same.
(D) All of the integers in the left-top to bottom-right diagonal are the same.
(E) All of the integers in the entire matrix are the same.
Sample Multiple-Choice Question 4

04. Consider the following two methods.

```java
/** Precondition: x > 0 and y > 0 */
public static int mango(int x, int y)
{
    int temp = papaya(x, y);
    return (x * y) / temp;
}

public static int papaya(int a, int b)
{
    int temp = a % b;
    if (temp == 0)
        return b;
    else
    {
        a = b;
        b = temp;
    }
    return papaya(a, b);
}
```

What value is returned as a result of the call `mango(45, 60)`?

(A) 15
(B) 90
(C) 180
(D) 1350
(E) 2700
Sample Multiple-Choice Question 5

05. Consider the following interface, classes and code segment.

```java
interface Language { public void greeting(); }

class German implements Language {
    public void greeting()
    {System.out.println("In German you say Guten Tag");}
}

class Dutch implements Language {
    public void greeting()
    {System.out.println("In Dutch you say Goeden Dag");}
}

class French implements Language {
    public void greeting()
    {System.out.println("In French you say Bonjour");}
}

ArrayList<Language> countries = new ArrayList<Language>();
countries.add(new German());
countries.add(new Dutch());
countries.add(new French());
for (Language country: countries)
    country.greeting();
```

What is printed as a result of executing the code segment?

(A) In German you say Guten Tag
   In Dutch you say Goeden Dag
   In French you say Bonjour

(B) Guten Tag
    Goeden Dag
    Bonjour

(C) German
    Dutch
    French

(D) A compile error message, because an abstract class was not used to implement
    the Language interface

(E) A compile error message, because the for loop structure does not access array elements properly.
Sample Multiple-Choice Question 6

06. Consider the following method.

```java
/** Precondition: n > 1 */
public static int method2126(int n)
{
    if (n == 1 || n == 2)
        return 1;
    else
        return n + method2126(n-1) + method2126(n-2);
}
```

What value is returned as a result of the call `method2126(5)`?

(A) 5  
(B) 10  
(C) 20  
(D) 36  
(E) 63
Consider the following program segment and List class.

```java
int[] x = {11,22,33,44,55,66,77,88,99};
List list = new List(x);
x[x.length] = 0;
System.out.println(list);

class List
{
    private int[] array;

    public List(int[] x)
    {
        array = x;
    }

    public String toString()
    {
        String temp = "[";
        for (int k = 0; k < array.length; k++)
        {
            temp = temp + array[k];
            if (k < array.length-1)
                temp = temp + ", ";
        }
        temp += "]";
        return temp;
    }
}
```

What is the output of executing the program segment?

(A) [0, 0, 0, 0, 0, 0, 0, 0, 0]
(B) [11, 22, 33, 44, 55, 66, 77, 88, 99]
(C) [11, 22, 33, 44, 55, 66, 77, 88, 0]
(D) [0, 22, 33, 44, 55, 66, 77, 88, 99]
(E) ArrayIndexOutOfBoundsException message
Sample Multiple-Choice Question 8

08. Consider the following method.

```java
/** Precondition: str is a nonempty string of upper-case letters
 * *
 * public static String oddString(String str)
 * {
 *     String temp = "";
 *     for (int k = 0; k < str.length(); k++)
 *     {
 *         String t = str.substring(k,k+1);
 *         System.out.print(t + " ");
 *         if (t != "A" && t != "E" && t != "I" && t != "O" && t != "U")
 *             temp += t;
 *     }
 *     return temp;
 * }
```

Which of the following is a correct Postcondition, based on the return of method `oddString`?

### (A) Postcondition: `oddString` returns `str`
(B) Postcondition: `oddString` returns `str` with all vowels removed
(C) Postcondition: `oddString` returns only the vowels in `str`
(D) Postcondition: `oddString` returns `null`
(E) It will generate a `StringOutOfBoundsException` error message
Sample Multiple-Choice Question 9

09. Consider the following code segment.

```java
for (int k = 1; k <= 100; k++)
{
    int n1 = (int) Math.random() * 2;
    int n2 = (int) Math.random() * 2;
    boolean bool = n1 == n2;
    System.out.print(bool + " ");
}
```

What is printed as a result of executing the code segment?

(A) The code segment always prints \textit{false}.
(B) The code segment always prints \textit{true}.
(C) The code segment prints \textit{true} roughly 50% of the time and prints \textit{false} roughly 50% of the time.
(D) The code segment prints both \textit{true} and \textit{false}, but \textit{true} more frequently.
(E) The code segment prints both \textit{true} and \textit{false}, but \textit{false} more frequently.
Sample Multiple-Choice Question 10

10. Consider the following two mystery methods. If both methods mystery1 and mystery2 are called with an identical integer array of \( n \) consecutive integers, starting at 1, like \( 1, 2, 3, 4, 5 \ldots, n \), what can be stated about the arrays at the conclusion of the execution?

```java
public static void mystery1(int[] x1) {
    for (int k = 1; k < x1.length; k++)
        x1[k] = x1[k] + x1[k-1];
}

public static void mystery2(int[] x2) {
    int increase = 2;
    for (int k = 1; k < x2.length; k++)
        { 
        x2[k] = x2[k-1] + increase;
        increase++;
        }
}
```

(A) Nothing can be stated with knowing the size of the arrays.
(B) Both arrays will be identical.
(C) Both arrays start with 1 and then they store different values.
(D) Both arrays store identical values, except for the first element.
(E) An ArrayIndexOutOfBoundsException error message
Sample Free-Response Question 1

Question 1.

This question involves converting numbers from Base16 to Base 10.

Part (a).

Write method `fromHexToBin`, which is described as follows. Method `fromHexToBin` converts the base-16 argument and converts the number to its base-2 equivalent. The base-2 number is returned. In completing method `fromHexToBin` you can use the following table, which details the unique relationship between base-16 and base-2 numbers. Every single base-16 digit corresponds to four base-2 digits.

<table>
<thead>
<tr>
<th>Base 16</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>a</th>
<th>b</th>
<th>c</th>
<th>d</th>
<th>e</th>
<th>f</th>
</tr>
</thead>
<tbody>
<tr>
<td>Base-2</td>
<td>0000</td>
<td>0001</td>
<td>0010</td>
<td>0011</td>
<td>0100</td>
<td>0101</td>
<td>0110</td>
<td>0111</td>
<td>1000</td>
<td>1001</td>
<td>1010</td>
<td>1011</td>
<td>1100</td>
<td>1101</td>
<td>1110</td>
<td>1111</td>
</tr>
</tbody>
</table>

Complete method `fromHexToBin` below.

```java
/** precondition: hexNum is a String of characters, which represents
 * a correct base-16 number.
 * postcondition: returns a String of characters, which represents
 * the base-2 number equivalent of hexNum.
 */
public static String fromHexToBin(String hexNum)
```

Part (b).

Write method `fromBinToDec`, which is described as follows. Method `fromBinToDec` converts the base-2 argument and converts the number to its base-10 equivalent. The base-10 number is returned.

Complete method `fromBinToDec` below.

```java
/** precondition: binNum is a String of characters, which represents
 * a correct base-2 number.
 * postcondition: returns an int value, which represents
 * the base-10 number equivalent of binNum.
 */
public static int fromBinToDec(String binNum)
```
Part (c).

Write method `fromHexToDec`, which is described as follows. Method `fromHexToDec` converts the base-16 argument and converts the number to its base-10 equivalent. The base-10 number is returned.

In writing method `fromHexToDec` you may call methods `fromHexToBin` and `fromBinToDec`, which are specified in Part (a) and Part(b), whether the methods are implemented correctly or not.

Complete method `fromHexToDec` below.

```java
/** precondition: hexNum is a String of characters, which represents
 * a correct base-16 number.
 * postcondition: returns an int value, which represents
 * the base-10 number equivalent of hexNum.
 */
public static int fromHexToDec(String hexNum)
```
Question 2.

This question involves examining Pascal's Triangle. Pascal's Triangle is a convenient way to compute the binomial coefficients of polynomials that are an expansion of \((x + y)^n\). Pascal's triangle can also be used for probability. The result of computing \((x + y)^5\) is shown below.

\[
(x + y)^5 = x^5 + 5x^4y + 10x^3y^2 + 10x^2y^3 + 5xy^4 + y^5
\]

Now consider the first 7 rows of Pascal's Triangle. Note that the numbers in row 6 correspond to the coefficients of the binomial expansion to the 5th power. Creating Pascal's triangle follows two rules:

1. The first and last number in each row are always 1.
2. All other numbers are the sum of two numbers in the previous row.

The triangle below demonstrates this pattern and more formally it can be stated that:

\[
\text{triangle[row][col]} = \text{triangle[row-1][col-1]} + \text{triangle[row-1][col]}
\]

\[
\begin{array}{cccccc}
1 \\
1 & 1 \\
1 & 2 & 1 \\
1 & 3 & 3 & 1 \\
1 & 4 & 6 & 4 & 1 \\
1 & 5 & 10 & 10 & 5 & 1 \\
1 & 6 & 15 & 20 & 15 & 6 & 1 \\
\end{array}
\]

Pascal's triangle can be viewed as a two-dimensional array. Since a two-dimensional array is in fact an array of arrays, it is not required that the array forms a square matrix. Each element of the array can be an array of a different size.

The number of rows in such an array, called \text{triangle.length}.

The number of columns in each row is \text{triangle[row].length}.
Consider the following incomplete declaration of a Pascal class.

The intent of the Pascal class is to create Pascal's Triangle.

```java
public class Pascal{
    private int n;
    private int[][] triangle;

    public Pascal(int n){
        /* implemented in part (a) */
    }

    public void makeTriangle(){
        /* implemented in part (b) */
    }

    public String toString(){
        /* not required */
    }
}
```

Part (a).

Complete constructor Pascal below.

```java
/**
 * precondition: n is >= 0
 * postcondition: space is allocated for a two-dimensional array, such
 * that row-0 has 1 element, row-1 has 2 elements, ...
 * and row-n has row+1 elements.
 */
public Pascal(int n)
```
Part (b).

Complete method `makeTriangle` below.

```java
/**
 * postcondition: Numbers are placed in n+1 rows of the triangle array
 * such that it represents Pascal's triangle.
 */
public makeTriangle()
```