Topics for Exam Review:

- Classes and Objects
- Arrays and Sorting
- Methods and Parameters Passing
- Program Modularity and Error Checking
- Run-Time Exceptions
- Defining a GUI Object using Inheritance
- Defining a Simple Class; Inheriting from class Object
- File Input and Wrapper Classes
- Dynamic vs. Static Structures: Linked Lists
- Throw an IllegalArgumentException
- JOptionPane (inputDialog, messageDialog)
- Read a String using a JOptionPane
- Character wrapper class
- Argument constructor
- Get and Set methods for each variable

Question 1.

Write a complete main method that does the following:

1. Takes 2 command line arguments, both integers.
2. If there are not 2 arguments, throw an IllegalArgumentException with an appropriate message.
3. Otherwise, output, to the console, the sum and product of those numbers, each on a different line.

For example,

```
C:>java Question2 12 5
12 + 5 = 17
12 * 5 = 60
```

Question 2.

Write the missing method below that will return the sum of a given row in the array.

```
public class Question4{
   public static void main(String args[]){
      int arr[][] = {{10, 45}, {2, 42, 67, 5}, {3, 21}};

      System.out.println(getSumofRow(arr, 1));  //prints the sum of row 1
   }
   private static int getSumOfRow (int [][] myArray, int row) {



   } // getSumOfRow
} //main
```

Question 3.

Write a Java program that will read a String using a JOptionPane, count the number of characters that are digits, and display that number in another JOptionPane. This may make use of the method in the Character wrapper class called isDigit().

Question 4.

Write a class called Book that contains the following information:

1. Instance variables for the title (String), author (String) and price (float) of the book.
2. A three-argument constructor to set each of the instance variable above. If the title or author is null or the price is less than 0, throw an IllegalArgumentException stating the argument that is not correct.
3. Get and Set methods for each variable.

```
public class Book {
```

1. **(25 points)** Write a Java application that accepts one String parameter from a *JOptionPane inputDialog* and prints to a *JOptionPane messageDialog* the number times *two* digits in a row occur. For example:

Input string:	messageDialog says
`Testing 12`	`Two digits happen 1 time.`
`Testing 123`	`Two digits happen 2 times.`
`123 Testing 12 23 234`	`Two digits happen 6 times.`

   Hint: Recall that there is a method in the Character wrapper class:

```
boolean Character.isDigit (char c)
```

2. (25 points) Write a boolean method that will return true if the sum of the diagonal (upper-left to lower-right) of a two-dimension array is equal to the sum of a given row:

```
diagonalEqualsRow ( int [][] myArray, int row)
```

If the array is not a square matrix of the row number is out of bounds, throw an IllegalArgumentException with an appropriate message.

For example:

myArray1

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

myArray2

1	3	3	2
4	2	5	6
1	**2**	**3**	**4**
7	8	9	4

myArray3

1	3	5	6	7
0	8	7	4	3

myArray4

1	3	3	2
4	2	5	6
1	**2**	**3**	**4**
7	8	9	4

```
diagonalEqualsRow (myArray1, 0) returns true
diagonalEqualsRow (myArray2, 2) returns true
diagonalEqualsRow (myArray3, 0) throw an exception – not a square matrix
diagonalEqualsRow (myArray1, 6) throw an exception – illegal row
```

3. (25 points) Create the following class. Read the entire question first!

Write a class `Distance` with the following specifications:

Class variables:
- a private static integer called INCHES_IN_A_FOOT with a value 12;
- a private static float called FEET_IN_A_METER with a value of 0.3048f;

Instance variables:
- Two integer instance variables: feet and inches;

Constructors:
- a no argument constructor to initialize a Distance with zero feet, zero inches.
- a two argument constructor that will accept two positive integers and for feet and inches. An "IllegalArgumentException" is thrown if the number of inches is more than 11.

Get/Set methods:
- get methods for the instance variables;
- set methods for instance variables.

Public methods:

1. a method add() that adds another Distance object to itself.

   That is,

```
w1 = new Distance(4,9);
w2 = new Distance (3,6);
w1.add(w2);              // w1 becomes 8 feet, 3 inches
```

2. a method metricDistance() that will return (as a float) the number of meters in the distance represented by the object.

   That is, for w1 above, w1.metricDistance() returns 1.4478f

4. (25 points) What will be printed by this program?

```
public class Question4 {
   static String s1,s2,s3,s4;
   public static void main (String[] args){

      s1 = new String("CAT");
      s2 = new String("DOG");
      s3 = new String("RAT");
      animal(s1,s2);
      System.out.println(mixThem(s2,s3));

   } //main
   private static void animal (String a1, String a2) {
      if(a1.equals(a2))
         System.out.println(a1);
      else
         System.out.println(a2);
   }
   private static String mixThem (String b1, String b2){
      String c1 = b1;
      b1 = b2;
      b2 = new String ("HAT");
      System.out.println(b1);
      System.out.println(b2);
      System.out.println(c1);
      System.out.println(s2);
      return b2;
   }

} //Question4
```