MIT AITI Mobile Application Development in Java
Nigeria, Summer 2012
Lab 05: Loops

Complete these problems and show your solutions to the instructors. Be prepared to explain your code. Create a new project in Eclipse named "lab05".

1. Create a Java file named `Lab05_1.java` that displays the first fifty *prime* numbers in five lines, each line contains 10 numbers. An integer greater than 1 is prime if its only positive divisor is 1 or itself. For example, 2, 3, 5, and 7 are prime but 4, 6, 8, and 9 are not prime. The output of your program should look like:

   ```
   The first 50 prime numbers are

   2 3 5 7 11 13 17 19 23 29

   31 37 41 43 47 53 59 61 67 71

   73 79 83 89 97 101 103 107 109 113

   127 131 137 139 149 151 157 163 167 173

   179 181 191 193 197 199 211 223 227 229
   ```

   You need to write a loop and test whether each new number is prime. Declare a variable `count` to store the number of primes encountered so far. If the number is prime, increment `count` by 1. When `count` is greater than 50, exit the loop.

   Hint: To test whether a number is prime, check if the number is divisible by 2, 3, 4, up to number/2. If a divisor is found, the number is not prime. For example, for the number 17, you need to test whether each of 2, 3, 4, 5, 6, 7, and 8 are divisors of 17. Since none are divisors, 17 is prime. If a number is not prime, once you find the first divisor, you should not keep checking for additional divisors

2. Use nested loops to print out each of the following patterns. Create a separate Java file for each pattern named `Lab05_2a.java, Lab05_2b.java, Lab05_2c.java, Lab05_2d.java,` and `Lab05_2e.java`.

a.
```
1
1 2
1 2 3
1 2 3 4
1 2 3 4 5
1 2 3 4 5 6
```

b.
```
1 2 3 4 5 6
1 2 3 4 5
1 2 3 4
1 2 3
1 2
1
```

c.
```
          1
        2 1
      3 2 1
    4 3 2 1
  5 4 3 2 1
6 5 4 3 2 1
```

d.
```
1 2 3 4 5 6
  1 2 3 4 5
    1 2 3 4
      1 2 3
        1 2
          1
```

e.
```
     1
    212
   32123
  4321234
 543212345
```

- (Extra Credit) Write nested loops that will print the following pattern:

```
                    1
                1   2   1
            1   2   4   2   1
        1   2   4   8   4   2   1
    1   2   4   8  16   8   4   2   1
        1   2   4   8  16  32  16   8   4   2   1
            1   2   4   8  16  32  64  32  16   8   4   2   1
        1   2   4   8  16  32  64 128  64  32  16   8   4   2   1
```

Reproduce the pattern exactly; note the spacing and how the digits align between different lines.