Lecture #3

3: Loops

Today’s Topics

- Loops
  - while
  - for
  - continue
  - break

Loops

Loops

```
static void main (String[] arguments) {
    System.out.println(“Rule #1”);
    System.out.println(“Rule #2”);
    System.out.println(“Rule #3”);
}
```

What if you want to do it for200 Rules?

Loops

Loop operators allow to loop through a block of code.

There are several loop operators in Java.

The *while* operator

```
while (condition) {
    statements
}
```

The *while* operator

```
int i = 0;
while (i < 3) {
    System.out.println("Rule #" + i);
    i = i+1;
}
```

Count carefully

Make sure that your loop has a chance to finish.

The *for* operator

```
for (initialization;condition;update){
     statements
}
```

The *for* operator

```
for (int i = 0; i < 3; i=i+1) {
    System.out.println("Rule #" + i);
}
```

Note: `i = i+1` may be replaced by `i++`

Branching Statements

break terminates a *for* or *while* loop

```
for (int i=0; i<100; i++) {
    if(i == 50)
          break;
    System.out.println("Rule #" + i);
}
```

Branching Statements

continue skips the current iteration of a loop and proceeds directly to the next iteration

```
fo   (int i=0; i<100; i++) {
    if(i == 50)
     continue;
    System.out.println("Rule #" + i);
}
```

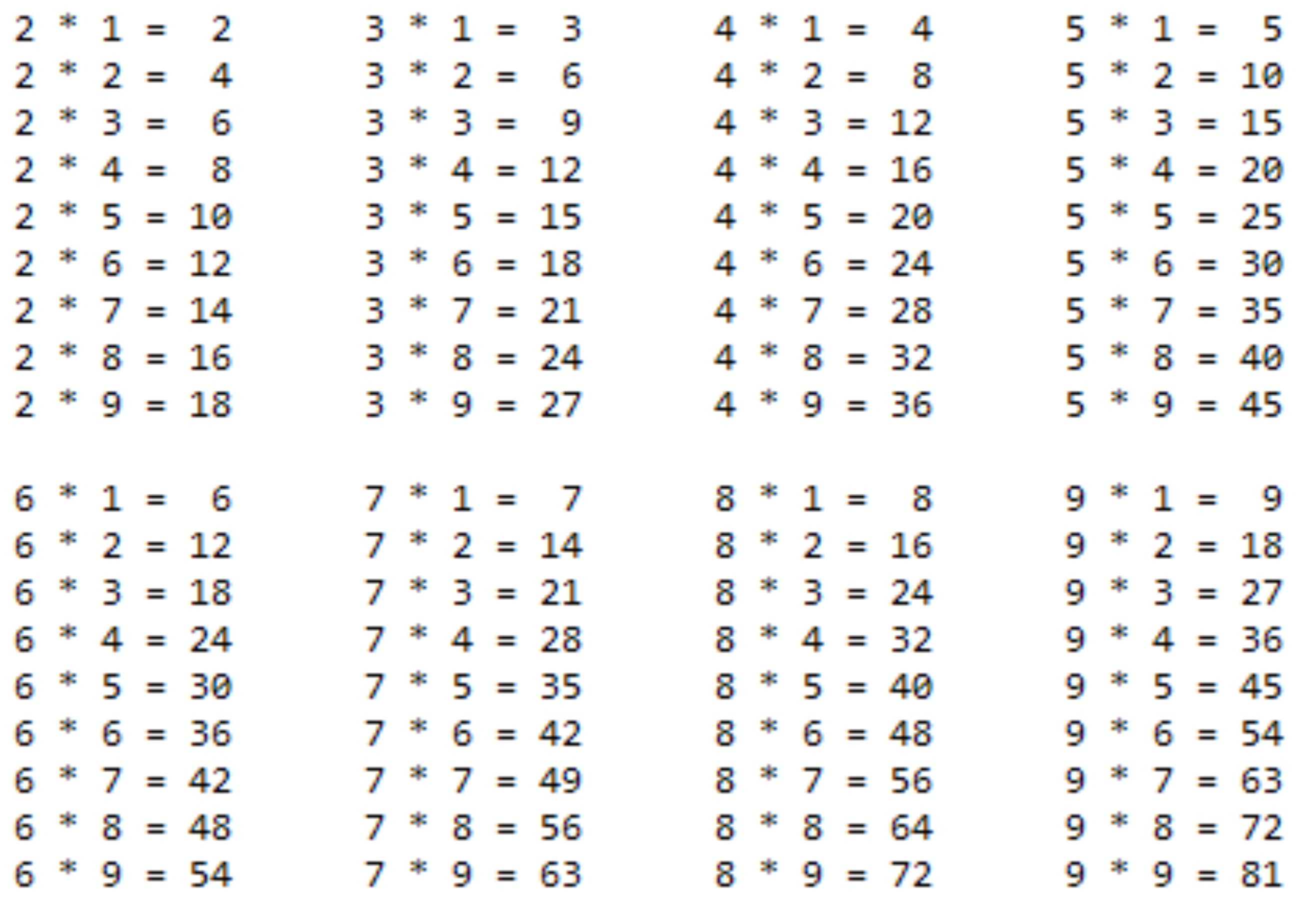
Embedded loops

```
for (int i = 0; i < 3; i++) {
    for (int j = 2; j < 4; j++) {
        System.out.println (i + " " + j);
    }
}
```

Scope of the variable defined in the initialization: respective *for* block

LAB2-1: Multiplication Table

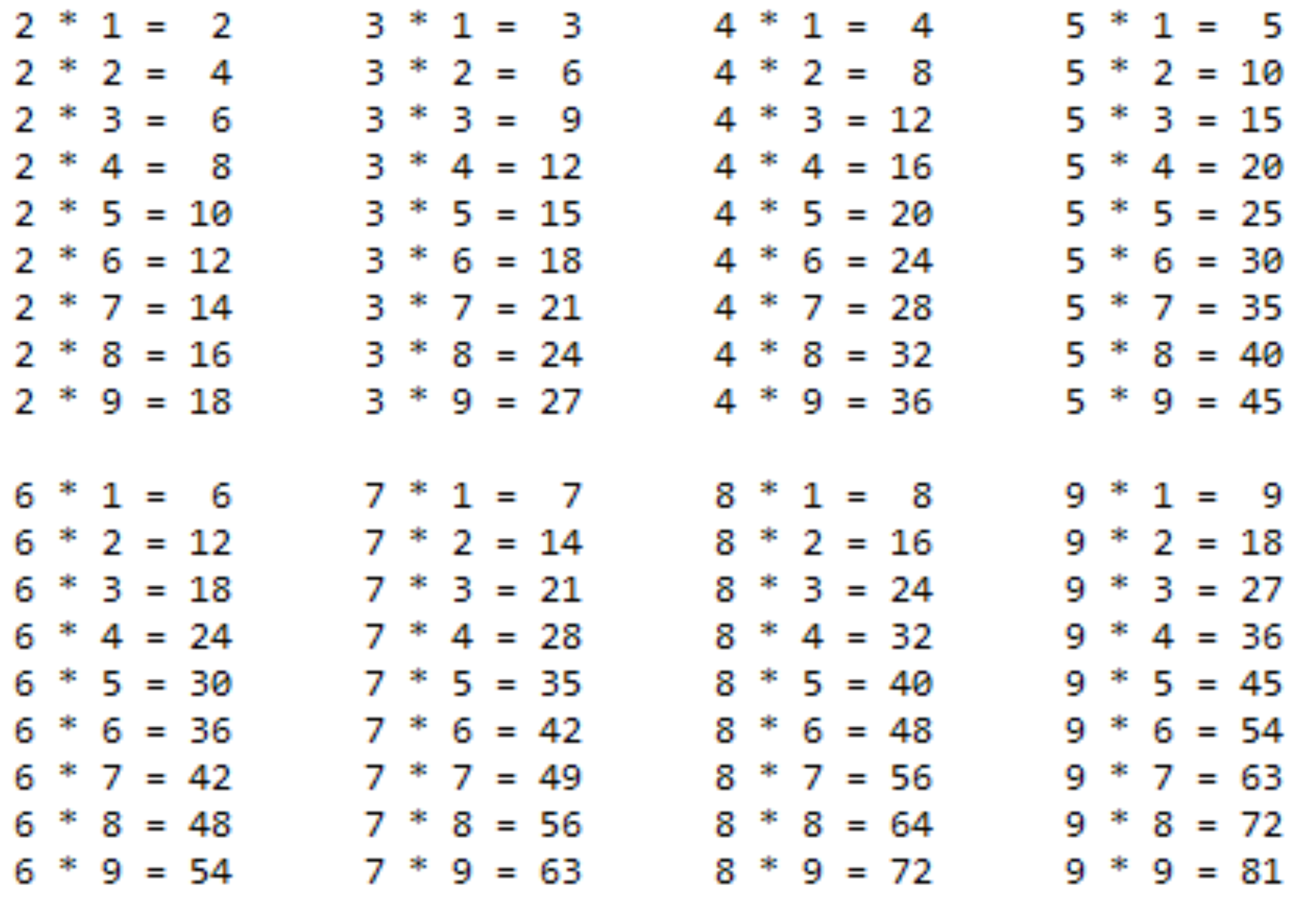
Let’s write a program which prints the multiplication table using the loop statement within another loop statement



```
2 * 1 =  2      3 * 1 =  3      4 * 1 =  4      5 * 1 =  5
2 * 2 =  4      3 * 2 =  6      4 * 2 =  8      5 * 2 = 10
2 * 3 =  6      3 * 3 =  9      4 * 3 = 12      5 * 3 = 15
2 * 4 =  8      3 * 4 = 12      4 * 4 = 16      5 * 4 = 20
2 * 5 = 10      3 * 5 = 15      4 * 5 = 20      5 * 5 = 25
2 * 6 = 12      3 * 6 = 18      4 * 6 = 24      5 * 6 = 30
2 * 7 = 14      3 * 7 = 21      4 * 7 = 28      5 * 7 = 35
2 * 8 = 16      3 * 8 = 24      4 * 8 = 32      5 * 8 = 40
2 * 9 = 18      3 * 9 = 27      4 * 9 = 36      5 * 9 = 45

6 * 1 =  6      7 * 1 =  7      8 * 1 =  8      9 * 1 =  9
6 * 2 = 12      7 * 2 = 14      8 * 2 = 16      9 * 2 = 18
6 * 3 = 18      7 * 3 = 21      8 * 3 = 24      9 * 3 = 27
6 * 4 = 24      7 * 4 = 28      8 * 4 = 32      9 * 4 = 36
6 * 5 = 30      7 * 5 = 35      8 * 5 = 40      9 * 5 = 45
6 * 6 = 36      7 * 6 = 42      8 * 6 = 48      9 * 6 = 54
6 * 7 = 42      7 * 7 = 49      8 * 7 = 56      9 * 7 = 63
6 * 8 = 48      7 * 8 = 56      8 * 8 = 64      9 * 8 = 72
6 * 9 = 54      7 * 9 = 63      8 * 9 = 72      9 * 9 = 81
```

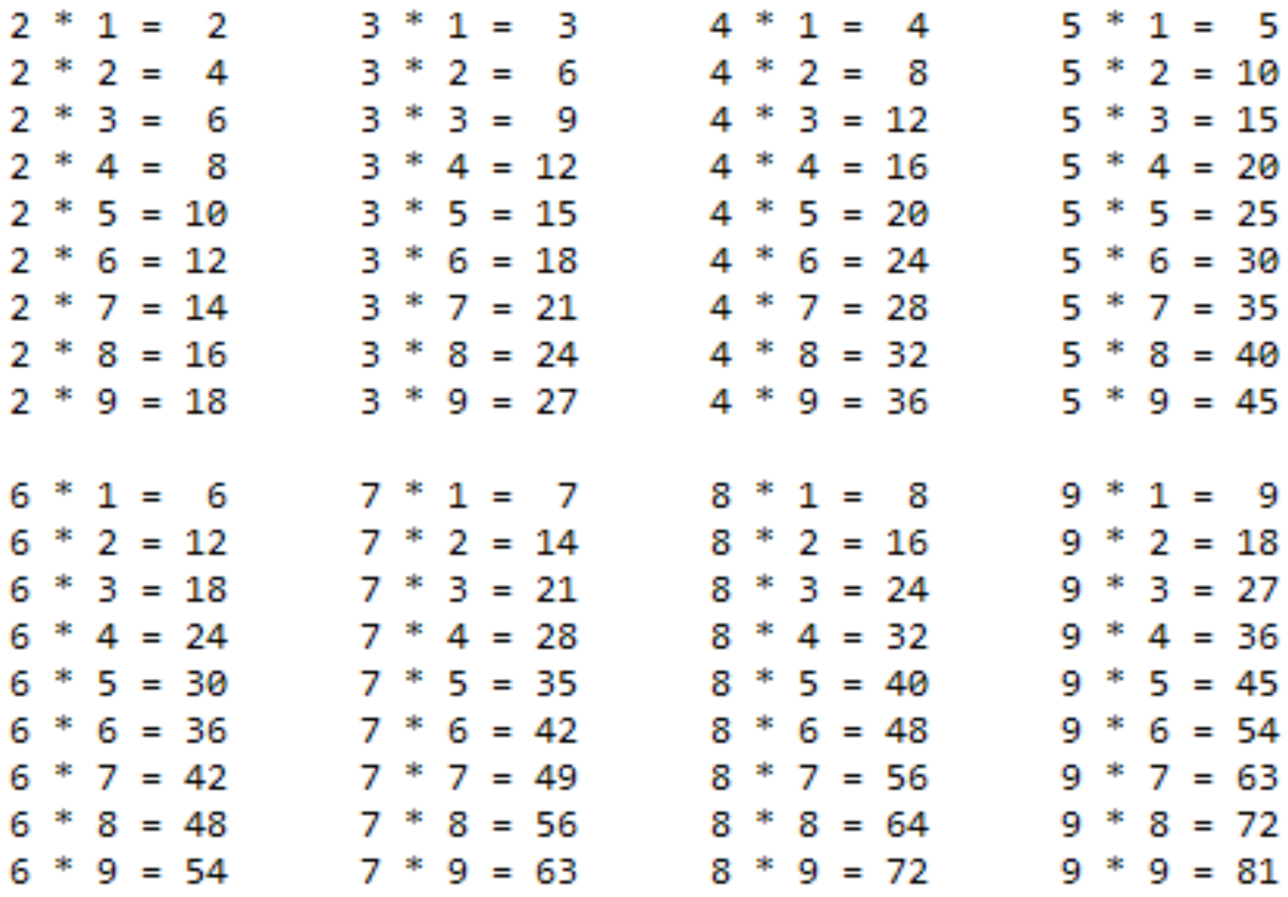
LAB2-1: Multiplication Table

Let’s write a program which prints the multiplication table using the loop statement within another loop statement



```
2 * 1 =  2      3 * 1 =  3      4 * 1 =  4      5 * 1 =  5
2 * 2 =  4      3 * 2 =  6      4 * 2 =  8      5 * 2 = 10
2 * 3 =  6      3 * 3 =  9      4 * 3 = 12      5 * 3 = 15
2 * 4 =  8      3 * 4 = 12      4 * 4 = 16      5 * 4 = 20
2 * 5 = 10      3 * 5 = 15      4 * 5 = 20      5 * 5 = 25
2 * 6 = 12      3 * 6 = 18      4 * 6 = 24      5 * 6 = 30
2 * 7 = 14      3 * 7 = 21      4 * 7 = 28      5 * 7 = 35
2 * 8 = 16      3 * 8 = 24      4 * 8 = 32      5 * 8 = 40
2 * 9 = 18      3 * 9 = 27      4 * 9 = 36      5 * 9 = 45

6 * 1 =  6      7 * 1 =  7      8 * 1 =  8      9 * 1 =  9
6 * 2 = 12      7 * 2 = 14      8 * 2 = 16      9 * 2 = 18
6 * 3 = 18      7 * 3 = 21      8 * 3 = 24      9 * 3 = 27
6 * 4 = 24      7 * 4 = 28      8 * 4 = 32      9 * 4 = 36
6 * 5 = 30      7 * 5 = 35      8 * 5 = 40      9 * 5 = 45
6 * 6 = 36      7 * 6 = 42      8 * 6 = 48      9 * 6 = 54
6 * 7 = 42      7 * 7 = 49      8 * 7 = 56      9 * 7 = 63
6 * 8 = 48      7 * 8 = 56      8 * 8 = 64      9 * 8 = 72
6 * 9 = 54      7 * 9 = 63      8 * 9 = 72      9 * 9 = 81
```

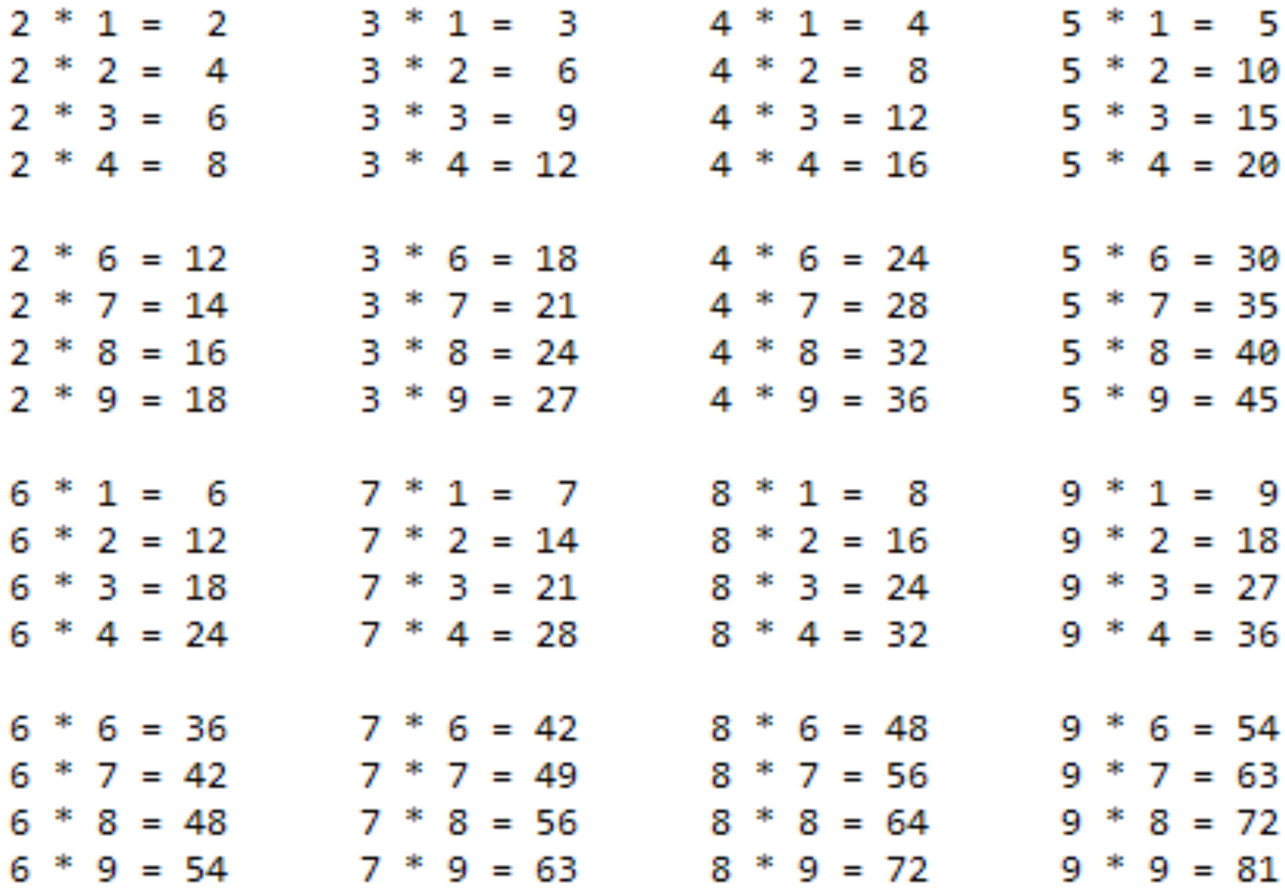
LAB2-1: Multiplication Table



```
2 * 1 =  2      3 * 1 =  3      4 * 1 =  4      5 * 1 =  5
2 * 2 =  4      3 * 2 =  6      4 * 2 =  8      5 * 2 = 10
2 * 3 =  6      3 * 3 =  9      4 * 3 = 12      5 * 3 = 15
2 * 4 =  8      3 * 4 = 12      4 * 4 = 16      5 * 4 = 20
2 * 5 = 10      3 * 5 = 15      4 * 5 = 20      5 * 5 = 25
2 * 6 = 12      3 * 6 = 18      4 * 6 = 24      5 * 6 = 30
2 * 7 = 14      3 * 7 = 21      4 * 7 = 28      5 * 7 = 35
2 * 8 = 16      3 * 8 = 24      4 * 8 = 32      5 * 8 = 40
2 * 9 = 18      3 * 9 = 27      4 * 9 = 36      5 * 9 = 45

6 * 1 =  6      7 * 1 =  7      8 * 1 =  8      9 * 1 =  9
6 * 2 = 12      7 * 2 = 14      8 * 2 = 16      9 * 2 = 18
6 * 3 = 18      7 * 3 = 21      8 * 3 = 24      9 * 3 = 27
6 * 4 = 24      7 * 4 = 28      8 * 4 = 32      9 * 4 = 36
6 * 5 = 30      7 * 5 = 35      8 * 5 = 40      9 * 5 = 45
6 * 6 = 36      7 * 6 = 42      8 * 6 = 48      9 * 6 = 54
6 * 7 = 42      7 * 7 = 49      8 * 7 = 56      9 * 7 = 63
6 * 8 = 48      7 * 8 = 56      8 * 8 = 64      9 * 8 = 72
6 * 9 = 54      7 * 9 = 63      8 * 9 = 72      9 * 9 = 81
```

LAB2-2: Multiplication Table

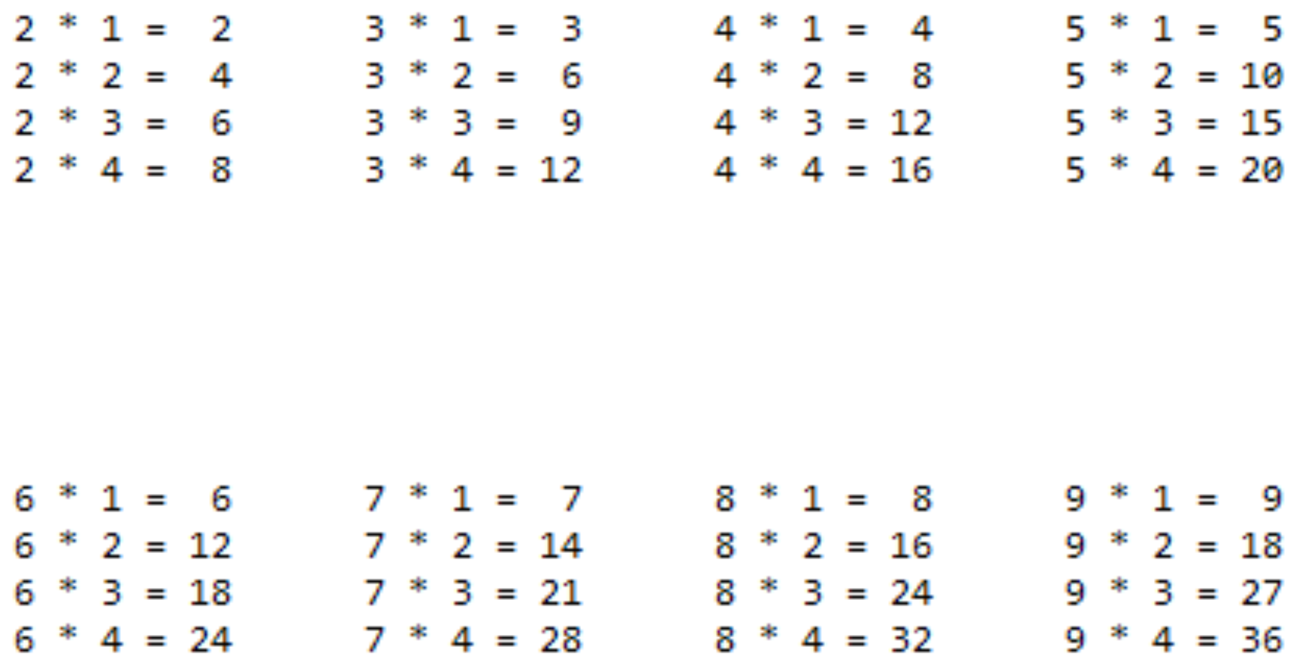
Let’s write a program which prints the multiplication table, but does not print the multiplication of each number by 5.



```
2 * 1 =  2      3 * 1 =  3      4 * 1 =  4      5 * 1 =  5
2 * 2 =  4      3 * 2 =  6      4 * 2 =  8      5 * 2 = 10
2 * 3 =  6      3 * 3 =  9      4 * 3 = 12      5 * 3 = 15
2 * 4 =  8      3 * 4 = 12      4 * 4 = 16      5 * 4 = 20

2 * 6 = 12      3 * 6 = 18      4 * 6 = 24      5 * 6 = 30
2 * 7 = 14      3 * 7 = 21      4 * 7 = 28      5 * 7 = 35
2 * 8 = 16      3 * 8 = 24      4 * 8 = 32      5 * 8 = 40
2 * 9 = 18      3 * 9 = 27      4 * 9 = 36      5 * 9 = 45

6 * 1 =  6      7 * 1 =  7      8 * 1 =  8      9 * 1 =  9
6 * 2 = 12      7 * 2 = 14      8 * 2 = 16      9 * 2 = 18
6 * 3 = 18      7 * 3 = 21      8 * 3 = 24      9 * 3 = 27
6 * 4 = 24      7 * 4 = 28      8 * 4 = 32      9 * 4 = 36

6 * 6 = 36      7 * 6 = 42      8 * 6 = 48      9 * 6 = 54
6 * 7 = 42      7 * 7 = 49      8 * 7 = 56      9 * 7 = 63
6 * 8 = 48      7 * 8 = 56      8 * 8 = 64      9 * 8 = 72
6 * 9 = 54      7 * 9 = 63      8 * 9 = 72      9 * 9 = 81
```

LAB2-3: Multiplication Table

Let’s write a program which prints the multiplication table, but does not print the multiplications of each number by 5, 6, 7, 8, and 9.



```
2 * 1 =  2      3 * 1 =  3      4 * 1 =  4      5 * 1 =  5
2 * 2 =  4      3 * 2 =  6      4 * 2 =  8      5 * 2 = 10
2 * 3 =  6      3 * 3 =  9      4 * 3 = 12      5 * 3 = 15
2 * 4 =  8      3 * 4 = 12      4 * 4 = 16      5 * 4 = 20






6 * 1 =  6      7 * 1 =  7      8 * 1 =  8      9 * 1 =  9
6 * 2 = 12      7 * 2 = 14      8 * 2 = 16      9 * 2 = 18
6 * 3 = 18      7 * 3 = 21      8 * 3 = 24      9 * 3 = 27
6 * 4 = 24      7 * 4 = 28      8 * 4 = 32      9 * 4 = 36
```

These slides are from:

- 6.092 Introduction to Programming in Java, January (IAP) 2010, MIT OpenCourseWare http://ocw.mit.edu
- Some of these slides are made by Seonah Lee