

## Pencil & Paper 06

---

**1. Write two nested for loops that display a multiplication table for the values 1 through 10 (HINT: the first line of the table will be "1 2 3 4 5 6 7 8 9 10" and the last line of the table will be "10 20 30 40 50 60 70 80 90 100").**

The following lines of code will produce a multiplication table:

```
for( int counterRow = 1; counterRow <= 10; counterRow++ ) {
    for( int counterColumn = 1; counterColumn <= 10; counterColumn++ ) {
        System.out.printf( "%5d", (counterColumn * counterRow) );
    }
    System.out.println();
}
```

**2. Write a for-each loop that displays the contents of the following array (HINT: Chapter 7, page 490) to the screen, one item per line.**

```
int [ ] numbers = {10, 20, 30, 40, 50};
```

The following lines of code will display the contents of the int array:

```
int[] numbers = { 10, 20, 30, 40, 50 };
for( int output : numbers ) {
    System.out.println( output );
}
```

**3. Write a for-each loop that displays the contents of the following array (HINT: Chapter 7, page 490) to the screen, one item per line.**

```
String [ ] messages = {"Hello","Goodbye", "Fare thee well"};
```

The following lines of code will display the contents of the String array:

```
String[] messages = { "Hello", "Goodbye", "Fare thee well" };
for( String output : messages ) {
    System.out.println( output );
}
```

**4. Write a heading and body for a method named *average* that returns *double* and takes two parameters of type *double* -- label the parameters *numberOne* and *numberTwo*. In the body of the method, write the Java code to compute and return the average of the two numbers.**

The following lines of code will compute the average of two double variables and return the answer as a data type double:

```
public double average( double numberOne, double numberTwo ) {
    return ((numberOne + numberTwo) / 2.0f);
}
```

**5. Explain the meaning of the Java keyword *this*.**

The *this* parameter allows a calling object to be named explicitly. If a local variable shares the same name as an instance variable then the instance variable is masked by default. To select the instance variable over the local variable the prefix *this* followed by a period and the variable name must be used (e.g. *this.sameNameVariable* instead of *sameNameVariable*).