



Reading Material

- Nested for loops: pp 119-136 of the text.
- Methods that take arguments and return values: sections 3.1 and 3.2, pp 156-179.

Exercises

1. Write a program `Triangle` to produce the output shown in 8 on page 149 of your book.
2. Write a program `Factorials` that print the factorials of the odd numbers between 1 and 10. Your program should contain a method that returns the factorial of an integer:

```
public static int factorial(int n)
```
3. Write a program `Payments` to calculate the monthly payments you would have to make over a given number of years to pay off a loan at a given interest rate compounded continuously. Your program should contain a method that takes the number of years t , the principal P , and the annual interest rate r as arguments. The desired value is given by the formula $(P e^{rt})/(t*12)$. Use `Math.exp()`. The signature of the method is:

```
public static double monthlyPayment(double years, double principal, double rate)
```

Use the method to generate a table of the monthly payments one would have to make for a \$20,000 loan over 5 years at interest rates varying from 5% to 8% in 0.25% increments.
4. Write a program `Fibonacci12` that computes and prints the first 12 Fibonacci numbers as described in Exercise 3 on page 148 of your textbook. Your program should consist of a method

```
public static fibonacci(int n)
```

that prints the first n Fibonacci numbers and gets called from the `main()` method.
5. Write a program `LetterArrow` that draws an arrow using the letter `b`. Your program should contain a method with the following signature:

```
public static void printArrow(String str)
```

which draws an arrow using a given string argument.

```
> java LetterArrow
b
  b
   b
    b
     b
```
6. Write the method `printGrid` that prints a grid of integers in column major order as described in Exercise 5 on p.208:

```
public static void printGrid(int rows, int columns)
```

Use it in a program `Grid46` that prints a 4x6 grid.
7. Write the following two methods:

```
public static int lastDigit(int number)
public static int nthDigit(int number, int n)
```

The methods should return the last digit of a number, and the digit that is n positions away from the last, respectively. When $n = 0$ we get the last digit, when $n=1$ we get the next-to-last digit, etc. Use these methods to write a program `Digits` which prints the last and next-to-last digits of all the numbers that are powers of 2 in the range 2^0 -- 2^{12} .
8. Write a method `nRandom` that takes an integer n as argument and uses `Math.random()` to print n uniform random values between 0.0 and 1.0, and then prints their average value.

Submission Instructions and Notes

- As usual, submit your commented source code and sample runs in a zip file to Moodle by the deadline.
- **Important:** We have changed the naming convention for your submitted files so it has a prefix indicating your section number. Your zip submissions should be named *s#_asst2_netid*, where *#* is your section number (between 1 and 9) and *netid* stands for your AUBnet user name. For example, if your AUBnetid is abc65 and you are in section 4, you should call your submission *s4_asst2_abc65*. The zip files will be processed automatically so please make sure you use this naming convention.
- For those of you using Windows, we will start using notepad++ to edit java programs. Please download from Moodle and install the proper version for your OS (32- or 64-bit). We will continue to use textedit on Mac OS.