



Reading Material

- Sections 1.4 – 1.5 of the text (pp 52-71).
- Chapter 2 of the textbook (pp 85-141).

Exercises

1. Write a program `Victory` that prints the output shown in Exercise 7 on page 79.
2. Write a program `Hexagons` that prints the output shown in Exercise 12 on page 81.
3. Write a program `Arrows` that draws the following sequence of forward and back arrows. You should write a method that draws a forward arrow, another one that draws a back arrow. These methods may be called from `main()`.



4. Solve Exercises 4, 12, and 15 on pages 142-145. No need to write complete programs for these exercises. Just write the solution in the text file that you are submitting.
5. Write a program `Position` that declares and initializes three `double` variables x_0 , v_0 , and t and prints the value of $x_0 + v_0t - gt^2/2$ where g is the constant 9.78033. Note: this value is the position (meters) after t seconds when an object is thrown straight up from initial position x_0 , at velocity v_0 (meters per second).
6. Write a program `Squares` with a `for` loop that produces the following output:
1 4 9 16 25 36 49 64 81 100
7. Write a program `LastDigits` that declares and initializes an integer variable n and prints the last and next-to-last digits of the number. For example, if n is 3672, your program should output:
The last digit of 3672 is 2.
The next to last digit of 3672 is 7.
8. Write a program `QuadraticRoots` that declares and initializes three `double` variables (a , b , c) and computes/prints the roots of the quadratic polynomial $ax^2 + bx + c$. Your program need to handle only the case when real roots exist, i.e. the values of a , b , c produce a positive discriminant $b^2 - 4ac > 0$.
Hint: `Math.sqrt()` is the built-in method that returns the square root of a number.

Submission Instructions

- Turn in a print out of your commented source code and sample runs for each program.
- You can show your sample runs by capturing and pasting portions of the command line window in a Word file and printing the file. Highlight the portion you want to copy with left mouse and press Enter (Ctrl-C does not work in the command window). Paste the copied portion in a Word file using Ctrl-V.
- Your submission must consist of a zip archive that contains `.java` files (one per program) and a file (`.txt` or `.doc`) containing sample runs and answers to exercises. No additional files should exist in the `.zip` archive.
- The name of the zip file must adhere to the following naming convention `asst1_<netid>`, where `<netid>` stands for you AUBnet user name. These zipped files will be processed automatically so please make sure you use this naming convention. The single zipped file must be uploaded to Moodle by the deadline.