2.1 Functions

Introduction to Programming in Java:  An Interdisciplinary Approach     ·     Robert Sedgewick and Kevin Wayne     ·     Copyright © 2002–2010     ·     2/17/11 9:58 PM

A Foundation for Programming

- any program you might want to write
- objects
- functions and modules
- graphics, sound, and image I/O
- arrays
- conditionals and loops
- Math
- text I/O
- primitive data types
- assignment statements
- build bigger programs and reuse code

Functions (Static Methods)

Java function.
- Takes zero or more input arguments.
- Returns one output value.
- Side effects (e.g., output to standard draw).

Applications.
- Scientists use mathematical functions to calculate formulas.
- Programmers use functions to build modular programs.
- You use functions for both.

Examples.
- Built-in functions: Math.random(), Math.abs(), Integer.parseInt().
- Our I/O libraries: StdIn.readInt(), StdDraw.line(), StdAudio.play().
- User-defined functions: main().
Anatomy of a Java Function

Java functions. Easy to write your own.

Flow of Control

Key point. Functions provide a new way to control the flow of execution.

What happens when a function is called:
- Control transfers to the function code.
- Argument variables are assigned the values given in the call.
- Function code is executed.
- Return value is assigned in place of the function name in calling code.
- Control transfers back to the calling code.

Note. This is known as "pass by value."

Scope

Scope (of a name). The code that can refer to that name.
Ex. A variable’s scope is code following the declaration in the block.

Best practice: declare variables to limit their scope.
Function Challenge 1a

Q. What happens when you compile and run the following code?

```java
public class Cubes1 {
    public static int cube(int i) {
        int j = i * i * i;
        return j;
    }
    public static void main(String[] args) {
        int N = Integer.parseInt(args[0]);
        for (int i = 1; i <= N; i++)
            StdOut.println(i + " " + cube(i));
    }
}
```

% javac Cubes1.java
% java Cubes1 6
1 1
2 8
3 27
4 64
5 125
6 216

Function Challenge 1b

Q. What happens when you compile and run the following code?

```java
public class Cubes2 {
    public static int cube(int i) {
        int j = i * i * i;
        return j;
    }
    public static void main(String[] args) {
        int N = Integer.parseInt(args[0]);
        for (int i = 1; i <= N; i++)
            StdOut.println(i + " " + cube(i));
    }
}
```

Function Challenge 1c

Q. What happens when you compile and run the following code?

```java
public class Cubes3 {
    public static int cube(int i) {
        i = i * i * i;
    }
    public static void main(String[] args) {
        int N = Integer.parseInt(args[0]);
        for (int i = 1; i <= N; i++)
            StdOut.println(i + " " + cube(i));
    }
}
```

Function Challenge 1d

Q. What happens when you compile and run the following code?

```java
public class Cubes4 {
    public static int cube(int i) {
        i = i * i * i;
        return i;
    }
    public static void main(String[] args) {
        int N = Integer.parseInt(args[0]);
        for (int i = 1; i <= N; i++)
            StdOut.println(i + " " + cube(i));
    }
}
```
Function Challenge 1e

Q. What happens when you compile and run the following code?

```java
public class Cubes5 {
    public static int cube(int i) {
        return i * i * i;
    }

    public static void main(String[] args) {
        int N = Integer.parseInt(args[0]);
        for (int i = 1; i <= N; i++)
            StdOut.println(i + " " + cube(i));
    }
}
```

Gaussian Distribution

Standard Gaussian distribution.
- "Bell curve."
- Basis of most statistical analysis in social and physical sciences.

Ex. 2000 SAT scores follow a Gaussian distribution with mean $\mu = 1019$, stddev $\sigma = 209$.

Mathematical functions. Use built-in functions when possible; build your own when not available.

```java
public class Gaussian {
    public static double phi(double x) {
        return Math.exp(-x*x / 2) / Math.sqrt(2 * Math.PI);
    }

    public static double phi(double x, double mu, double sigma) {
        return phi((x - mu) / sigma) / sigma;
    }
}
```

Overloading. Functions with different signatures are different.
Multiple arguments. Functions can take any number of arguments.
Calling other functions. Functions can call other functions.
Gaussian Cumulative Distribution Function

**Goal.** Compute Gaussian cdf $\Phi(z)$.

**Challenge.** No "closed form" expression and not in Java library.

**Bottom line.** 1,000 years of mathematical formulas at your fingertips.

\[
\Phi(z) = \int_{-\infty}^{z} \phi(x) \, dx = \int_{-\infty}^{z} \frac{1}{\sqrt{2\pi}} e^{-x^2/2} \, dx
\]

Java function for $\Phi(z)$

```java
public class Gaussian {
    public static double phi(double x)
        // as before
    ...

    public static double Phi(double z, double mu, double sigma) {
        return Phi((z - mu) / sigma);
    }
}
```

accurate with absolute error less than $8 \times 10^{-16}$

\[
\Phi(z, \mu, \sigma) = \int_{-\infty}^{z} \phi(z, \mu, \sigma) = \Phi((z - \mu) / \sigma)
\]

SAT Scores

**Q.** NCAA requires at least 820 for Division I athletes. What fraction of test takers in 2000 do not qualify?

**A.** $\Phi(820, 1019, 209) \approx 0.17051$. [approximately 17%]

Gaussian Distribution

**Q.** Why relevant in mathematics?

**A.** Central limit theorem: under very general conditions, average of a set of random variables tends to the Gaussian distribution.

**Q.** Why relevant in the sciences?

**A.** Models a wide range of natural phenomena and random processes.
  - Weights of humans, heights of trees in a forest.
  - SAT scores, investment returns.

**Caveat.**

"Everybody believes in the exponential law of errors: the experimenters, because they think it can be proved by mathematics; and the mathematicians, because they believe it has been established by observation."

— M. Lippman in a letter to H. Poincaré
Building Functions

Functions enable you to build a new layer of abstraction.
- Takes you beyond pre-packaged libraries.
- You build the tools you need: Gaussian.phi(), ...

Process.
- Step 1: identify a useful feature.
- Step 2: implement it.
- Step 3: use it.
- Step 3': re-use it in any of your programs.

Digital Audio

Crash Course in Sound

Sound. Perception of the vibration of molecules in our eardrums.

Concert A. Sine wave, scaled to oscillate at 440Hz.
Other notes. 12 notes on chromatic scale, divided logarithmically.

<table>
<thead>
<tr>
<th>note</th>
<th>frequency</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>0 440.00</td>
</tr>
<tr>
<td>A# or B♭</td>
<td>1 466.16</td>
</tr>
<tr>
<td>B</td>
<td>2 493.88</td>
</tr>
<tr>
<td>C</td>
<td>3 523.25</td>
</tr>
<tr>
<td>Cs or D♭</td>
<td>4 554.37</td>
</tr>
<tr>
<td>D</td>
<td>5 587.33</td>
</tr>
<tr>
<td>D# or E♭</td>
<td>6 622.25</td>
</tr>
<tr>
<td>E</td>
<td>7 659.26</td>
</tr>
<tr>
<td>F</td>
<td>8 698.46</td>
</tr>
<tr>
<td>Fs or G♭</td>
<td>9 739.99</td>
</tr>
<tr>
<td>G</td>
<td>10 783.99</td>
</tr>
<tr>
<td>G# or A♭</td>
<td>11 830.61</td>
</tr>
<tr>
<td>A</td>
<td>12 880.00</td>
</tr>
</tbody>
</table>

Notes, numbers, and waves

Digital Audio

Sampling. Represent curve by sampling it at regular intervals.
Musical Tone Function

Musical tone. Create a music tone of a given frequency and duration.

```java
public static double[] tone(double hz, double seconds) {
    int SAMPLE_RATE = 44100;
    int N = (int) (seconds * SAMPLE_RATE);
    double[] a = new double[N + 1];
    for (int i = 0; i <= N; i++) {
        a[i] = Math.sin(2 * Math.PI * i * hz / SAMPLE_RATE);
    }
    return a;
}
```

Remark. Can use arrays as function return value and/or argument.

Standard audio. Library for playing digital audio.

```java
public class StdAudio
{
    void play(String file) { ... }
    void play(double[] a) { ... }
    void save(String file, double[] a) { ... }
    double[] read(String file) { ... }
}
```

Concert A. Play concert A for 1.5 seconds using StdAudio.

```java
double[] a = tone(440, 1.5);
StdAudio.play(a);
```

Harmonics

Concert A with harmonics. Obtain richer sound by adding tones one octave above and below concert A.

880 Hz 220 Hz 440 Hz

Concert A with harmonics

Harmonics

Concert A with harmonics

Harmonics
Harmonics

Play that tune. Read in pitches and durations from standard input, and play using standard audio.

```java
public static void main(String[] args) {
    while (!StdIn.isEmpty()) {
        int pitch = StdIn.readInt();
        double duration = StdIn.readDouble();
        double[] a = note(pitch, duration);
        StdAudio.play(a);
    }
}
```

% more elise.txt  % java PlayThatTune < elise.txt
7 .125
6 .125
7 .125
6 .125
7 .125
2 .125
5 .125
3 .125
0 .25