### 2.1 Functions

**A Foundation for Programming**

- **Objects**
  - Functions and modules
  - Graphics, sound, and image I/O
  - Arrays
  - Conditionals and loops
  - Math
  - Text I/O
  - Primitive data types
  - Assignment statements

- **Any program you might want to write**
  - 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**
- Our I/O libraries: `Stdin.readInt()`, `StdDraw.line()`, `StdAudio.play()`.
- User-defined functions: `main()`.

**Flow of Control**

**Key point.** Functions provide a new way to control the flow of execution.
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 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 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));
    }
}
```
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.

Example: 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.

Overloading. Functions with different signatures are different.
- Multiple arguments.
- Functions can take any number of arguments.
- Calling other functions. Functions can call other functions.

Java Function for $\phi(x)$

```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);
    }
}
```

Gaussian Cumulative Distribution Function

Goal. Compute Gaussian cdf $\Phi(x)$.
Challenge. No "closed form" expression and not in Java library.

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

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

A. \( \Phi(820.0, 1019.0, 209.0) \approx 0.17051 \). [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.pdf(), ...

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.

Extra Slides

Function Examples

<table>
<thead>
<tr>
<th>absolute value of an int value</th>
<th>public static int abs(int x)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>{ if (x &lt; 0) return -x;</td>
</tr>
<tr>
<td></td>
<td>else return x; }</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>absolute value of a double value</th>
<th>public static double abs(double x)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>{ if (x &lt; 0.0) return -x;</td>
</tr>
<tr>
<td></td>
<td>else return x; }</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>primality test</th>
<th>public static boolean isPrime(int x)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>{ if (x &lt; 2) return false;</td>
</tr>
<tr>
<td></td>
<td>for (int i = 2; i &lt;= N/2; i++)</td>
</tr>
<tr>
<td></td>
<td>if (N % i == 0) return false;</td>
</tr>
<tr>
<td></td>
<td>return true; }</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>hypotenuse of a right triangle</th>
<th>public static double hypotenuse(double a, double b)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>{ return Math.sqrt(a^2 + b^2); }</td>
</tr>
</tbody>
</table>