Procedural Programming: Your First Programs

Data Structures and Algorithms in Java

Outline

Outline

Outline

The Java Language

Programming in Java

Application Programming Interface (API)

Input and Output

Errors in a Program

Outline

Outline

Outline

Outline

Outline

The Java Language

The Java Language

General-purpose, high-level, object-oriented programming language

The Java Language

General-purpose, high-level, object-oriented programming language

Key features:

The Java Language

General-purpose, high-level, object-oriented programming language

Key features:

- Write once, run anywhere

The Java Language

General-purpose, high-level, object-oriented programming language

Key features:

- Write once, run anywhere
- Relatively fast

The Java Language

General-purpose, high-level, object-oriented programming language

Key features:

- Write once, run anywhere
- Relatively fast
- Robust

The Java Language

General-purpose, high-level, object-oriented programming language

Key features:

- Write once, run anywhere
- Relatively fast
- Robust
- Secure

Programming in Java

Programming in Java

Step 1: Create/edit the program (eg, `Program.java`)

Programming in Java

Step 1: Create/edit the program (eg, `Program.java`)

Step 2: Compile the program

```
$ _
```

Programming in Java

Step 1: Create/edit the program (eg, `Program.java`)

Step 2: Compile the program

```
$ javac -d out src/Program.java
```

Programming in Java

Step 1: Create/edit the program (eg, `Program.java`)

Step 2: Compile the program

```
$ javac -d out src/Program.java
$ _
```

Programming in Java

Step 1: Create/edit the program (eg, `Program.java`)

Step 2: Compile the program

```
$ javac -d out src/Program.java
$ _
```

Step 3: Run the generated program `out/Program.class`

```
$ _
```

Programming in Java

Step 1: Create/edit the program (eg, `Program.java`)

Step 2: Compile the program

```
$ javac -d out src/Program.java
$ _
```

Step 3: Run the generated program `out/Program.class`

```
$ java Program
```

Programming in Java

Step 1: Create/edit the program (eg, `Program.java`)

Step 2: Compile the program

```
$ javac -d out src/Program.java
$ _
```

Step 3: Run the generated program `out/Program.class`

```
$ java Program
<program output>
$ _
```

Programming in Java

Step 1: Create/edit the program (eg, `Program.java`)

Step 2: Compile the program

```
$ javac -d out src/Program.java
$ _
```

Step 3: Run the generated program `out/Program.class`

```
$ java Program
<program output>
$ _
```

Repeat steps 1 — 3 until program output matches expected

Programming in Java

Programming in Java

Program (`HelloWorld.java`):

Programming in Java

Program (`HelloWorld.java`):

- Standard output: the message `"Hello, World"`

Programming in Java

Program (`HelloWorld.java`):

- Standard output: the message `"Hello, World"`

```
$ _
```

Programming in Java

Program (`HelloWorld.java`):

- Standard output: the message `"Hello, World"`

```
$ javac -d out src/HelloWorld.java
```

Programming in Java

Program (`HelloWorld.java`):

- Standard output: the message `"Hello, World"`

```
$ javac -d out src/HelloWorld.java
$ _
```

Programming in Java

Program (`HelloWorld.java`):

- Standard output: the message `"Hello, World"`

```
$ javac -d out src/HelloWorld.java
$ java HelloWorld
```

Programming in Java

Program (`HelloWorld.java`):

- Standard output: the message `"Hello, World"`

```
$ javac -d out src/HelloWorld.java
$ java HelloWorld
Hello, World
$ _
```

Programming in Java

Programming in Java

```java
// Writes the message "Hello, World" as standard output.

import stdlib.StdOut;

public class HelloWorld {
    // Entry point.
    public static void main(String[] args) {
        StdOut.println("Hello, World");
    }
}
```

Application Programming Interface (API)

Application Programming Interface (API)

API is a set of protocols that allows different software applications to communicate with one another

Application Programming Interface (API)

API is a set of protocols that allows different software applications to communicate with one another

Example (API for the `stdlib.StdOut` library):

Application Programming Interface (API)

API is a set of protocols that allows different software applications to communicate with one another

Example (API for the `stdlib.StdOut` library):

`class StdOut`	
`static void println(Object x)`	writes object `x` followed by newline to standard output
`static void print(Object x)`	writes object `x` to standard output

Input and Output

Input and Output

Input and Output

Input types:

Input and Output

Input types:

- Command-line input

Input and Output

Input types:

- Command-line input
- Standard input

Input and Output

Input types:

- Command-line input
- Standard input
- File input

Input and Output

Input types:

- Command-line input
- Standard input
- File input

Output types:

Input and Output

Input types:

- Command-line input
- Standard input
- File input

Output types:

- Standard output

Input and Output

Input types:

- Command-line input
- Standard input
- File input

Output types:

- Standard output
- File output

Input and Output

Input and Output ▶ Command-line Input

Input and Output ▶ Command-line Input

Command-line inputs (aka arguments) are strings listed next to the program name during execution

```
$ _
```

Input and Output ▶ Command-line Input

Command-line inputs (aka arguments) are strings listed next to the program name during execution

```
$ java Program input1 input2 input3 ...
```

Input and Output ▶ Command-line Input

Command-line inputs (aka arguments) are strings listed next to the program name during execution

```
$ java Program input1 input2 input3 ...
```

The inputs are accessed within the entry-point function as `args[0]`, `args[1]`, `args[2]`, ...

Example:

Input and Output ▶ Command-line Input

Example:

```
$ _
```

Input and Output ▶ Command-line Input

Example:

```
$ java Program Galileo "Isaac Newton" Einstein
```

Input and Output ▶ Command-line Input

Example:

```
$ java Program Galileo "Isaac Newton" Einstein
```

args[0]	args[1]	args[2]

Input and Output ▶ Command-line Input

Example:

```
$ java Program Galileo "Isaac Newton" Einstein
```

args[0]	args[1]	args[2]
Galileo		

Input and Output ▶ Command-line Input

Example:

```
$ java Program Galileo "Isaac Newton" Einstein
```

args[0]	args[1]	args[2]
Galileo	Isaac Newton	

Input and Output ▶ Command-line Input

Example:

```
$ java Program Galileo "Isaac Newton" Einstein
```

args[0]	args[1]	args[2]
Galileo	Isaac Newton	Einstein

Input and Output ▶ Command-line Input

Program (`UseArgument.java`):

Program (`UseArgument.java`):

- Command-line input: a name

Program (`UseArgument.java`):

- Command-line input: a name
- Standard output: a message containing the name

Input and Output ▶ Command-line Input

Program (`UseArgument.java`):

- Command-line input: a name
- Standard output: a message containing the name

```
$ _
```

Input and Output ▶ Command-line Input

Program (`UseArgument.java`):

- Command-line input: a name
- Standard output: a message containing the name

```
$ javac -d out src/UseArgument.java
```

Input and Output ▶ Command-line Input

Program (`UseArgument.java`):

- Command-line input: a name
- Standard output: a message containing the name

```
$ javac -d out src/UseArgument.java
$ _
```

Input and Output ▶ Command-line Input

Program (`UseArgument.java`):

- Command-line input: a name
- Standard output: a message containing the name

```
$ javac -d out src/UseArgument.java
$ java UseArgument Alice
```

Input and Output ▶ Command-line Input

Program (`UseArgument.java`):

- Command-line input: a name
- Standard output: a message containing the name

```
$ javac -d out src/UseArgument.java
$ java UseArgument Alice
Hi, Alice. How are you?
$ _
```

Input and Output ▶ Command-line Input

Program (`UseArgument.java`):

- Command-line input: a name
- Standard output: a message containing the name

```
$ javac -d out src/UseArgument.java
$ java UseArgument Alice
Hi, Alice. How are you?
$ java UseArgument Bob
```

Input and Output ▶ Command-line Input

Program (`UseArgument.java`):

- Command-line input: a name
- Standard output: a message containing the name

```
$ javac -d out src/UseArgument.java
$ java UseArgument Alice
Hi, Alice. How are you?
$ java UseArgument Bob
Hi, Bob. How are you?
$ _
```

Input and Output ▶ Command-line Input

Program (`UseArgument.java`):

- Command-line input: a name
- Standard output: a message containing the name

```
$ javac -d out src/UseArgument.java
$ java UseArgument Alice
Hi, Alice. How are you?
$ java UseArgument Bob
Hi, Bob. How are you?
$ java UseArgument Carol
```

Input and Output ▶ Command-line Input

Program (`UseArgument.java`):

- Command-line input: a name
- Standard output: a message containing the name

```
$ javac -d out src/UseArgument.java
$ java UseArgument Alice
Hi, Alice. How are you?
$ java UseArgument Bob
Hi, Bob. How are you?
$ java UseArgument Carol
Hi, Carol. How are you?
$ _
```

Input and Output ▶ Command-line Input

Input and Output ▶ Command-line Input

```
 1 // Receives a name as command-line input; and writes a message containing
 2 // that name as standard output.
 3
 4 import stdlib.StdOut;
 5
 6 public class UseArgument {
 7     // Entry point.
 8     public static void main(String[] args) {
 9         StdOut.print("Hi, ");
10         StdOut.print(args[0]);
11         StdOut.println(". How are you?");
12     }
13 }
```

Errors in a Program

Errors in a Program ▶ Compile-time Errors

Errors in a Program ▶ Compile-time Errors

Compile-time errors are identified and reported by Java when it compiles a program

Errors in a Program ▶ Compile-time Errors

Compile-time errors are identified and reported by Java when it compiles a program

Example:

```java
// Receives a name as command-line input; and writes a message containing
// that name as standard output.

import stdlib.StdOut;

public class UseArgument {
    // Entry point.
    public static void main(String[] args) {
        StdOut.print("Hi, ");
        StdOut.print(args[0);
        StdOut.println(". How are you?");
    }
}
```

```
$ _
```

Errors in a Program ▶ Compile-time Errors

Compile-time errors are identified and reported by Java when it compiles a program

Example:

```
 1 // Receives a name as command-line input; and writes a message containing
 2 // that name as standard output.
 3
 4 import stdlib.StdOut;
 5
 6 public class UseArgument {
 7     // Entry point.
 8     public static void main(String[] args) {
 9         StdOut.print("Hi, ");
10         StdOut.print(args[0);
11         StdOut.println(". How are you?");
12     }
13 }
```

```
$ javac -d out src/UseArgument.java
```

Errors in a Program ▶ Compile-time Errors

Compile-time errors are identified and reported by Java when it compiles a program

Example:

```java
// Receives a name as command-line input; and writes a message containing
// that name as standard output.

import stdlib.StdOut;

public class UseArgument {
    // Entry point.
    public static void main(String[] args) {
        StdOut.print("Hi, ");
        StdOut.print(args[0);
        StdOut.println(". How are you?");
    }
}
```

```
$ javac -d out src/UseArgument.java
UseArgument.java:10: error: ']' expected
        StdOut.print(args[0);
                           ^
1 error
$ _
```

Errors in a Program

Errors in a Program ▶ Run-time Errors

Errors in a Program ▶ Run-time Errors

Run-time errors are identified and reported by Java when it runs a program

Errors in a Program ▶ Run-time Errors

Run-time errors are identified and reported by Java when it runs a program

Example:

```java
// Receives a name as command-line input; and writes a message containing
// that name as standard output.

import stdlib.StdOut;

public class UseArgument {
    // Entry point.
    public static void main(String[] args) {
        StdOut.print("Hi, ");
        StdOut.print(args[0]);
        StdOut.println(". How are you?");
    }
}
```

```
$ _
```

Errors in a Program ▶ Run-time Errors

Run-time errors are identified and reported by Java when it runs a program

Example:

```
 1 // Receives a name as command-line input; and writes a message containing
 2 // that name as standard output.
 3
 4 import stdlib.StdOut;
 5
 6 public class UseArgument {
 7     // Entry point.
 8     public static void main(String[] args) {
 9         StdOut.print("Hi, ");
10         StdOut.print(args[0]);
11         StdOut.println(". How are you?");
12     }
13 }
```

```
$ javac -d out src/UseArgument.java
```

Errors in a Program ▶ Run-time Errors

Run-time errors are identified and reported by Java when it runs a program

Example:

```java
// Receives a name as command-line input; and writes a message containing
// that name as standard output.

import stdlib.StdOut;

public class UseArgument {
    // Entry point.
    public static void main(String[] args) {
        StdOut.print("Hi, ");
        StdOut.print(args[0]);
        StdOut.println(". How are you?");
    }
}
```

```
$ javac -d out src/UseArgument.java
$ _
```

Errors in a Program ▶ Run-time Errors

Run-time errors are identified and reported by Java when it runs a program

Example:

```java
// Receives a name as command-line input; and writes a message containing
// that name as standard output.

import stdlib.StdOut;

public class UseArgument {
    // Entry point.
    public static void main(String[] args) {
        StdOut.print("Hi, ");
        StdOut.print(args[0]);
        StdOut.println(". How are you?");
    }
}
```

```
$ javac -d out src/UseArgument.java
$ java UseArgument
```

Errors in a Program ▶ Run-time Errors

Run-time errors are identified and reported by Java when it runs a program

Example:

```java
// Receives a name as command-line input; and writes a message containing
// that name as standard output.

import stdlib.StdOut;

public class UseArgument {
    // Entry point.
    public static void main(String[] args) {
        StdOut.print("Hi, ");
        StdOut.print(args[0]);
        StdOut.println(". How are you?");
    }
}
```

```
$ javac -d out src/UseArgument.java
$ java UseArgument
Hi, Exception in thread "main" java.lang.ArrayIndexOutOfBoundsException: Index 0 out of bounds for length 0
 at UseArgument.main(UseArgument.java:10)
$ _
```

Errors in a Program

Errors in a Program ▶ Logic Errors

Errors in a Program ▶ Logic Errors

Logic errors are neither identified nor reported by Java, but produce unintended output

Errors in a Program ▶ Logic Errors

Logic errors are neither identified nor reported by Java, but produce unintended output

Example:

```java
// Receives a name as command-line input; and writes a message containing
// that name as standard output.

import stdlib.StdOut;

public class UseArgument {
    // Entry point.
    public static void main(String[] args) {
        StdOut.print("Hi, ");
        StdOut.print(args[0]);
        StdOut.print(". How are you?");
    }
}
```

```
$ _
```

Errors in a Program ▶ Logic Errors

Logic errors are neither identified nor reported by Java, but produce unintended output

Example:

```java
// Receives a name as command-line input; and writes a message containing
// that name as standard output.

import stdlib.StdOut;

public class UseArgument {
    // Entry point.
    public static void main(String[] args) {
        StdOut.print("Hi, ");
        StdOut.print(args[0]);
        StdOut.print(". How are you?");
    }
}
```

```
$ javac -d out src/UseArgument.java
```

Errors in a Program ▶ Logic Errors

Logic errors are neither identified nor reported by Java, but produce unintended output

Example:

```java
// Receives a name as command-line input; and writes a message containing
// that name as standard output.

import stdlib.StdOut;

public class UseArgument {
    // Entry point.
    public static void main(String[] args) {
        StdOut.print("Hi, ");
        StdOut.print(args[0]);
        StdOut.print(". How are you?");
    }
}
```

```
$ javac -d out src/UseArgument.java
$ _
```

Errors in a Program ▶ Logic Errors

Logic errors are neither identified nor reported by Java, but produce unintended output

Example:

```java
// Receives a name as command-line input; and writes a message containing
// that name as standard output.

import stdlib.StdOut;

public class UseArgument {
    // Entry point.
    public static void main(String[] args) {
        StdOut.print("Hi, ");
        StdOut.print(args[0]);
        StdOut.print(". How are you?");
    }
}
```

```
$ javac -d out src/UseArgument.java
$ java UseArgument Alice
```

Errors in a Program ▶ Logic Errors

Logic errors are neither identified nor reported by Java, but produce unintended output

Example:

```java
// Receives a name as command-line input; and writes a message containing
// that name as standard output.

import stdlib.StdOut;

public class UseArgument {
    // Entry point.
    public static void main(String[] args) {
        StdOut.print("Hi, ");
        StdOut.print(args[0]);
        StdOut.print(". How are you?");
    }
}
```

```
$ javac -d out src/UseArgument.java
$ java UseArgument Alice
Hi, Alice. How are you?$ _
```