

# **Data Structures and Algorithms in Java**

Procedural Programming: Your First Programs

## Outline

- ① Java
- ② Programming in Java
- ③ Application Programming Interface (API)
- ④ Input and Output
- ⑤ Errors in a Program



# Java

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

# Java

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

## Key features

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



## 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

```
>_ ~/workspace/dsaj
```

```
$ _
```



# Programming in Java

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

Step 2: Compile the program

```
>_ ~/workspace/dsaj
```

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

# Programming in Java

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

Step 2: Compile the program

```
>_ ~/workspace/dsaj
$ javac -d out src/Program.java
$ -
```

# Programming in Java

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

Step 2: Compile the program

```
>_ ~/workspace/dsaj
$ javac -d out src/Program.java
$ _
```

Step 3: Run the program

```
>_ ~/workspace/dsaj
$ _
```

# Programming in Java

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

Step 2: Compile the program

```
>_ ~/workspace/dsaj
$ javac -d out src/Program.java
$ -
```

Step 3: Run the program

```
>_ ~/workspace/dsaj
$ java Program
```

# Programming in Java

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

Step 2: Compile the program

```
>_ ~/workspace/dsaj
$ javac -d out src/Program.java
$ _
```

Step 3: Run the program

```
>_ ~/workspace/dsaj
$ java Program
<program output>
$ _
```

# Programming in Java

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

Step 2: Compile the program

```
>_ ~/workspace/dsaj
$ javac -d out src/Program.java
$ _
```

Step 3: Run the program

```
>_ ~/workspace/dsaj
$ java Program
<program output>
$ _
```

Repeat steps 1 – 3 until program output matches expected



## Programming in Java · Program Template

<> Program.java

```
// Package statement.
[package dsa;]

// Import statements.
...

// Outer class definition.
public class Program [implements <name>] {
    // Class/instance variable declarations.
    ...

    // Constructor definitions.
    ...

    // Method definitions.
    ...

    // Inner class definitions.
    ...

    // Function definitions.
    ...
}
```





## Programming in Java · Example (Hello World)

 HelloWorld.java

Standard output | the message "Hello, World"

## Programming in Java · Example (Hello World)

 HelloWorld.java

Standard output | the message "Hello, World"

>\_ ~/workspace/dsaj

\$ \_

## Programming in Java · Example (Hello World)

 HelloWorld.java

Standard output | the message "Hello, World"

>\_ ~/workspace/dsaj

\$ javac -d out src/HelloWorld.java

## Programming in Java · Example (Hello World)

 HelloWorld.java

Standard output | the message "Hello, World"

>\_ ~/workspace/dsaj

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

## Programming in Java · Example (Hello World)

 HelloWorld.java

Standard output | the message "Hello, World"

>\_ ~/workspace/dsaj

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

## Programming in Java · Example (Hello World)

HelloWorld.java

Standard output | the message "Hello, World"

>\_ ~/workspace/dsaj

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





## Programming in Java · Example (Hello World)

<> HelloWorld.java

```
1 // Writes the message "Hello, World" to standard output.
2
3 import stdlib.Stdout;
4
5 public class HelloWorld {
6     // Entry point.
7     public static void main(String[] args) {
8         StdOut.println("Hello, World");
9     }
10 }
```

## Application Programming Interface (API)

## Application Programming Interface (API)

API is a set of rules or protocols that allow different software applications to communicate with each other

## Application Programming Interface (API)

API is a set of rules or protocols that allow different software applications to communicate with each other

The API for a library describes the behavior of its functions

## Application Programming Interface (API)

API is a set of rules or protocols that allow different software applications to communicate with each other

The API for a library describes the behavior of its functions

### Example

stdlib.Stdout

<code>static void println(Object x)</code>	prints an object and a newline to standard output
<code>static void print(Object x)</code>	prints an object to standard output

## Input and Output

## Input and Output



## 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

- Standard output
- File output



## Input and Output · Command-line Input

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

```
>_ ~/workspace/dsaj
```

```
$ 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

```
>_ ~/workspace/dsaj  
$ java Program input1 input2 input3 ...
```

The inputs are accessed within the entry-point (ie, `public static void main(String[] args) {...}`) function as `args[0]`, `args[1]`, `args[2]`, ...

## Input and Output · Command-line Input

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

```
>_ ~/workspace/dsaj  
$ java Program input1 input2 input3 ...
```

The inputs are accessed within the entry-point (ie, `public static void main(String[] args) {...}`) function as `args[0]`, `args[1]`, `args[2]`, ...

### Example

```
>_ ~/workspace/dsaj  
$ java Program Galileo "Isaac Newton" Einstein
```

<code>args[0]</code>	<code>args[1]</code>	<code>args[2]</code>
<code>"Galileo"</code>	<code>"Isaac Newton"</code>	<code>"Einstein"</code>

**Input and Output** · Example (User Input)

## Input and Output · Example (User Input)

UseArgument.java

Command-line input

a name

Standard output

a message containing the name

## Input and Output · Example (User Input)

UseArgument.java

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

>\_ ~/workspace/dsaj

\$ \_



## Input and Output · Example (User Input)

UseArgument.java

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

>\_ ~/workspace/dsaj

\$ javac -d out src/UseArgument.java

## Input and Output · Example (User Input)

UseArgument.java

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

>\_ ~/workspace/dsaj

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

```
$ -
```

## Input and Output · Example (User Input)

UseArgument.java

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

>\_ ~/workspace/dsaj

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

## Input and Output · Example (User Input)

UseArgument.java

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

>\_ ~/workspace/dsaj

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

## Input and Output · Example (User Input)

UseArgument.java

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

>\_ ~/workspace/dsaj

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

## Input and Output · Example (User Input)

UseArgument.java

Command-line input

a name

Standard output

a message containing the name

>\_ ~/workspace/dsaj

```
$ 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 · Example (User Input)

UseArgument.java

Command-line input

a name

Standard output

a message containing the name

>\_ ~/workspace/dsaj

```
$ 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 · Example (User Input)

UseArgument.java

Command-line input

a name

Standard output

a message containing the name

>\_ ~/workspace/dsaj

```
$ 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** · Example (User Input)

## Input and Output · Example (User Input)

</> UseArgument.java

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

## Errors in a Program · Compile-time Errors

## Errors in a Program · Compile-time Errors

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

## Errors in a Program · Compile-time Errors

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

### Example

</> UseArgument.java

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

## Errors in a Program · Compile-time Errors

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

### Example

```
</> UseArgument.java
```

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

```
>_ ~/workspace/dsaj
```

```
$ _
```

## Errors in a Program · Compile-time Errors

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

### Example

```
</> UseArgument.java
```

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

```
>_ ~/workspace/dsaj
```

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

## Errors in a Program · Compile-time Errors

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

### Example

```
</> UseArgument.java
```

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

```
>_ ~/workspace/dsaj
```

```
$ javac -d out src/UseArgument.java
UseArgument.java:9: error: ']' expected
    StdOut.print(args[0]);
                    ^
1 error
$ _
```



## 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

<> UseArgument.java

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

## Errors in a Program · Run-time Errors

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

### Example

```
<> UseArgument.java
```

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

```
>_ ~/workspace/dsaj
```

```
$ _
```

## Errors in a Program · Run-time Errors

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

### Example

```
<> UseArgument.java
```

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

```
>_ ~/workspace/dsaj
```

```
$ 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

```
<> UseArgument.java
```

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

```
>_ ~/workspace/dsaj
```

```
$ 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

```
<> UseArgument.java
```

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

```
>_ ~/workspace/dsaj
```

```
$ 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

```
<> UseArgument.java
```

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

```
>_ ~/workspace/dsaj
```

```
$ 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:9)
$ _
```



## 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

</> UseArgument.java

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

## Errors in a Program · Logic Errors

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

### Example

```
<> UseArgument.java
```

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

```
>_ ~/workspace/dsaj
```

```
$ _
```

## Errors in a Program · Logic Errors

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

### Example

```
<> UseArgument.java
```

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

```
>_ ~/workspace/dsaj
```

```
$ 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

```
<> UseArgument.java
```

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

```
>_ ~/workspace/dsaj
```

```
$ 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

```
<> UseArgument.java
```

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

```
>_ ~/workspace/dsaj
```

```
$ 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

```
<> UseArgument.java
```

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

```
>_ ~/workspace/dsaj
```

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