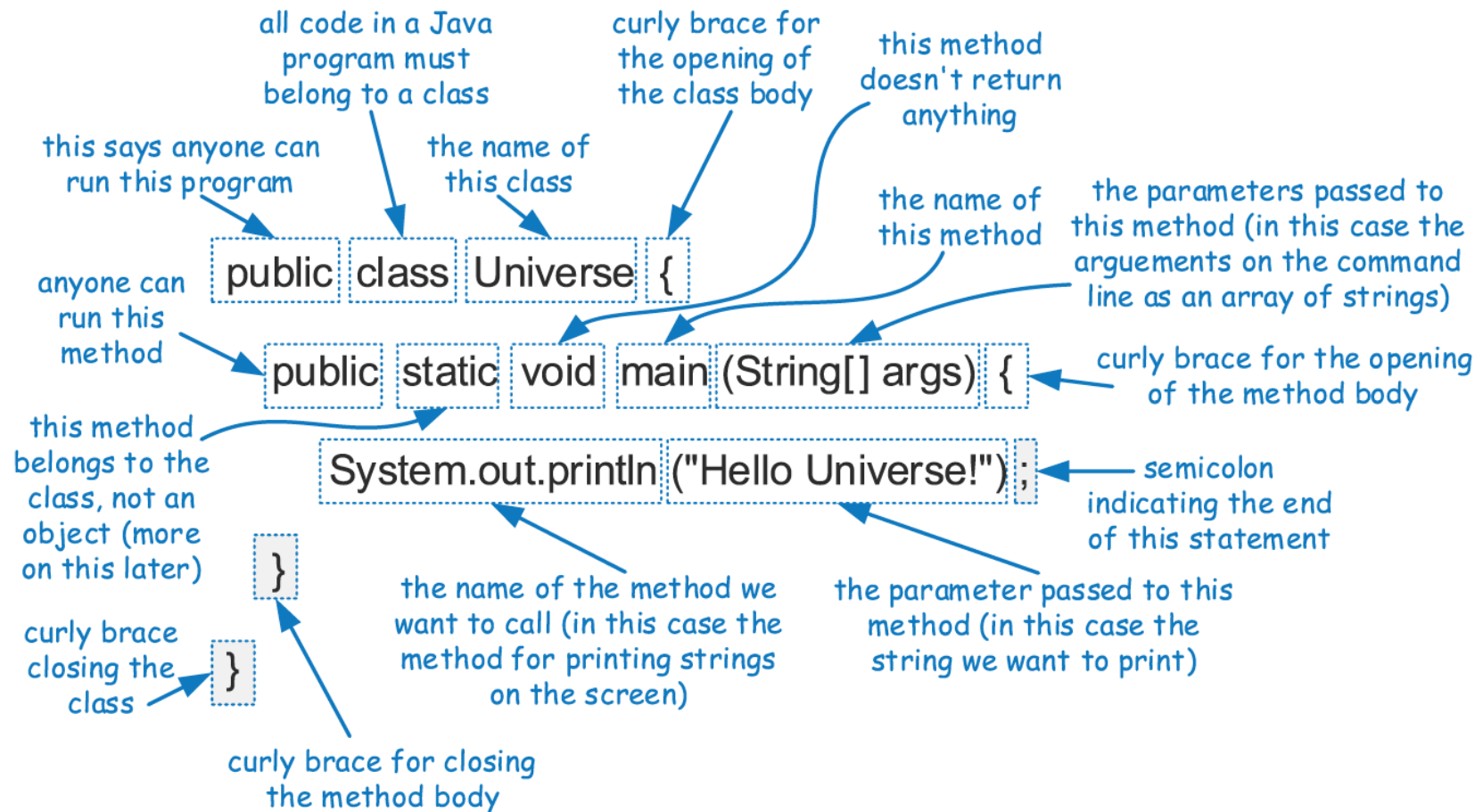


Data Structures and Algorithms

Chapter 1

Java Basics

Sample Program



Java Basics

Components of a Java Program

- In Java, executable statements are placed in functions, known as **methods**, that belong to class definitions.
- The static method named **main** is the first method to be executed when running a Java program.
- Any set of statements between the braces “{” and “}” define a program block.
- Examples:
 - SampleProgram1.java
 - SampleProgram2.java

Java Basics

Primitive (or Base) Types

- Primitive types:
 - **byte**: 8-bit signed 2's complement integer; from -128 to 127, inclusive
 - **short**: 16-bit signed 2's complement integer; from -32768 to 32767, inclusive
 - **int**: 32-bit signed 2's complement integer; from -2147483648 to 2147483647, inclusive
 - **long**: 64-bit signed 2's complement integer; from -9223372036854775808 to 9223372036854775807, inclusive
 - **char**: 16-bit Unicode character; from '\u0000' to '\uffff' inclusive, that is, from 0 to 65535
 - **float**: single-precision, 32-bit floating point number (IEEE 754-1985)
 - **double**: double-precision, 64-bit floating point number (IEEE 754-1985)
 - **boolean**: true or false

Java Basics

Reference Types

- Reference types: class types, interface types, array types.
- Values of a reference type: references to objects
- A reference variable stores the location (i.e., memory address) of an object.
- Example:
 - `PrimitiveReference.java`

Java Basics

Access Control Modifier

- Also called *access level modifier* or *visibility modifier*.
- Declared for classes, variables, and methods.

Modifier	Access Level			
	Class	Package	Subclass	World
public	Y	Y	Y	Y
protected	Y	Y	Y	N
no modifier	Y	Y	N	N
private	Y	N	N	N

Java Basics

When a New Object is Created

- Use the *new* operator and the constructor.
- Memory is dynamically allocated.
- Instance variables are initialized .
- The *new* operator returns the *reference* to the new object.
- The reference is assigned to an instance variable (a reference to the object).

Java Basics

Static Modifier

- Specified for variables or methods of a class.
- They belong to the class not to an instance of the class.
- Example:
 - Car.java
 - TestCar.java

Java Basics

Wrapper Class

Primitive Type	Wrapper Class
boolean	Boolean
char	Character
byte	Byte
short	Short
int	Integer
long	Long
float	Float
double	Double

Java Basics

Wrapper Class

- Autoboxing and autounboxing

```
public class BoxingTest {  
    public static void main(String[ ] args) {  
        Integer a = 1024; // primitive value 1024 is boxed into an object  
        System.out.println("a is " + a.intValue());  
        int b = a + 10; // object a is unboxed to primitive type  
        System.out.println("b is " + b);  
    }  
}
```

Java Basics

Casting

- Narrowing vs. widening type conversion

```
double x = 3.14
```

```
int a = (int)x; // narrowing conversion from  
               // double to int
```

```
double y = a;  // widening conversion from int  
               //to double
```

Java Basics

Control Flow

- if statements

```
if (booleanExpression)  
    trueBody  
else  
    falseBody
```

Java Basics

Control Flow

- if statements

```
if (firstBooleanExpression)  
    firstBody  
else if (secondBooleanExpression)  
    secondBody  
else  
    thirdBody
```

Java Basics

Control Flow

- **switch** statements

```
switch (var) {  
    case value1:    // var == value1  
        do something;  
        break;  
    case value2:    // var == value2  
        do something;  
        break;  
    . . .  
    default        // none of the above  
        do something  
}
```

Java Basics

Control Flow

- **for** loops

for (initialization; booleanCondition; increment)
 loopBody

Meaning:

```
{  
    initialization;  
    while (booleanCondition) {  
        loopBody;  
        increment;  
    }  
}
```

Java Basics

Control Flow

- **while** loops

```
while (booleanExpression)  
    loopBody
```

- **do-while** loops

```
do  
    loopBody  
while (booleanExpression)
```

- Example: `ControlFlowExamples.java`

Java Basics

Arrays

- Declaration

```
int [ ] intArray; // array of integers
```

```
double [ ] doubleArray; // array of doubles
```

```
Char [ ] charArray; // array of characters
```

```
String [ ] stringArray; // array of strings
```

- Allocate memory, and initialize

```
intArray = new int [5];
```

```
IntArray[0] = 10;
```

```
IntArray[1] = 20;
```

```
IntArray[2] = 30;
```

```
IntArray[3] = 40;
```

```
IntArray[4] = 50;
```

Java Basics

Arrays

- Declare and allocate memory

```
Int [ ] intArray = new int[10];
```

- Shortcut

```
Int [ ] intArray = {10, 20, 30, 40, 50};
```

- Example: `ArrayExample.java`

Java Basics

Simple I/O

- Read from standard input and write to standard output example:
 - SimpleIOTest1.java
 - SimpleIOTest2.java
- Read from a text file and write to a text file:
 - SimpleIOTest3.java
 - There are other ways

References

- M.T. Goodrich, R. Tamassia, and M.H. Goldwasser, “Data Structures and Algorithms in Java,” Sixth Edition, Wiley, 2014.