ITSE322 Modern Programing Language (Advanced Java)

Lecturer:	Dr. Harmain Mohammed
eMail:	h.harmain@it.uot.edu.ly
Course Site:	classhub

Java Programming: From Problem Analysis to Program Design

Introduction to Objects
and
the String Class

Objectives

- Learn about objects and reference variables
- Explore how to use predefined methods in a program
- Become familiar with the `class String`

Java Variables

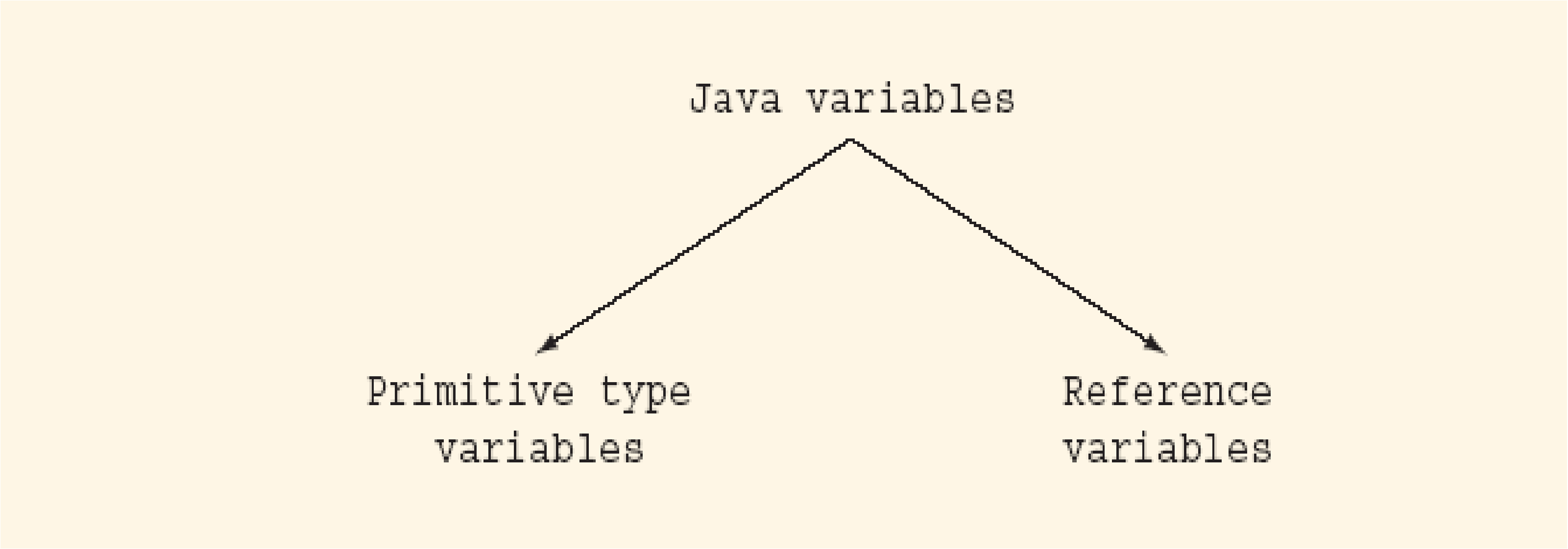


FIGURE 3-6 Java variables

Object and Reference Variables

```
int x;          //Primitive variable
String str;     //Reference variable
x = 45;         //x stores simple value
str = new String("Java Programming“);
```

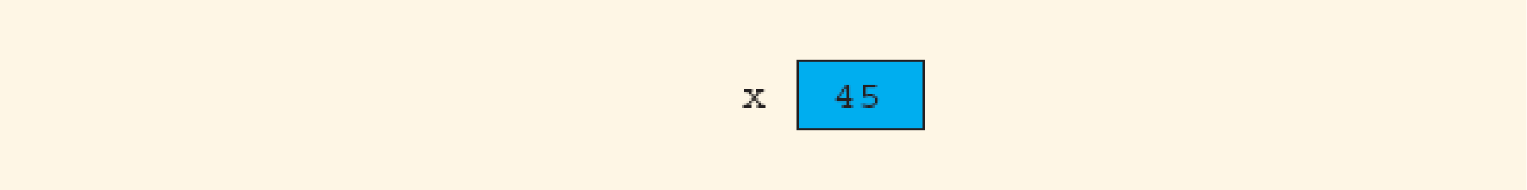


FIGURE 3-1 Variable x and its data

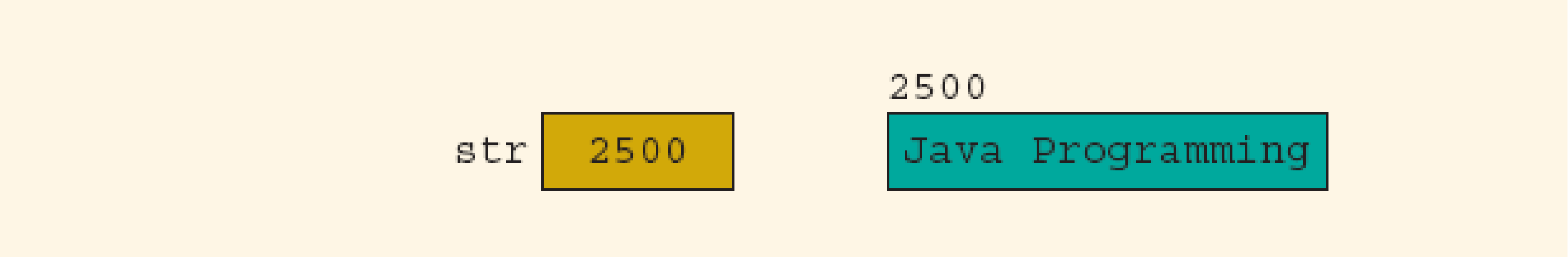


FIGURE 3-2 Variable str and the data it points to

Object and Reference Variables (continued)

```
str = new String(“Hello there!”);
```

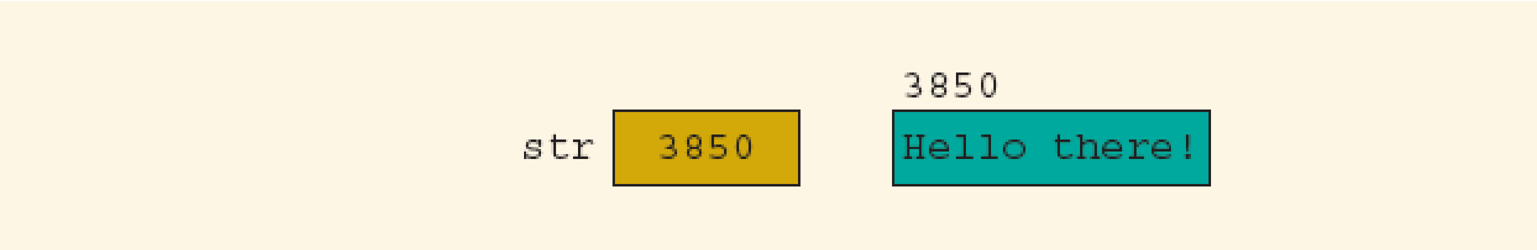


FIGURE 3-4 Variable str, its value, and the object str

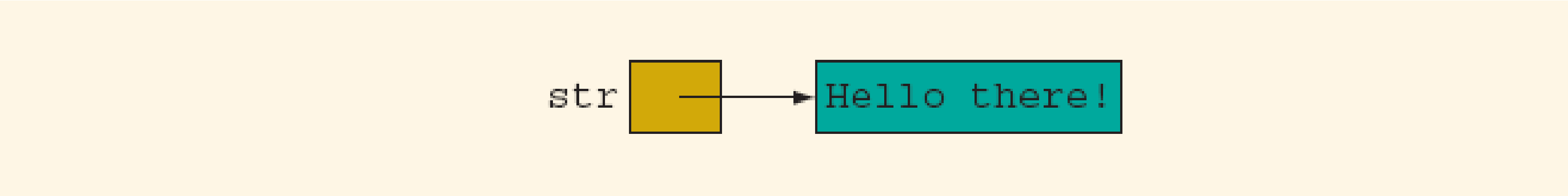


FIGURE 3-5 Variable str and the object str

Objects and Reference Variables (continued)

- Primitive type variables directly store data into their memory space
- Reference variables store the address of the object containing the data
- An object is an instance of a `class` and the operator `new` is used to instantiate an object

Using Predefined Classes and Methods in a Program

- Java Library: collection of packages
- Package: collection of classes
- Class: contains data and methods
- Method: set of instructions

Using Predefined Classes and Methods in a Program (continued)

- To use a method you must know:
  - Name of the package containing class (`java.lang`)
  - Name of the class containing the method (`Math`)
  - Name of the method - (`pow`), and its parameters
  - `Math.pow(x, y)` means x^y

Using Predefined Classes and Methods in a Program (continued)

- Example method call

```
import java.lang.Math; //imports package
int x = Math.pow(2, 3);//calls power method
                        // in class Math
```

- Dot (.) Operator is used to access the method in the class

The `class String`

- String variables are reference variables
- Given:

```
String name;
```

  - The following Statements do the same thing

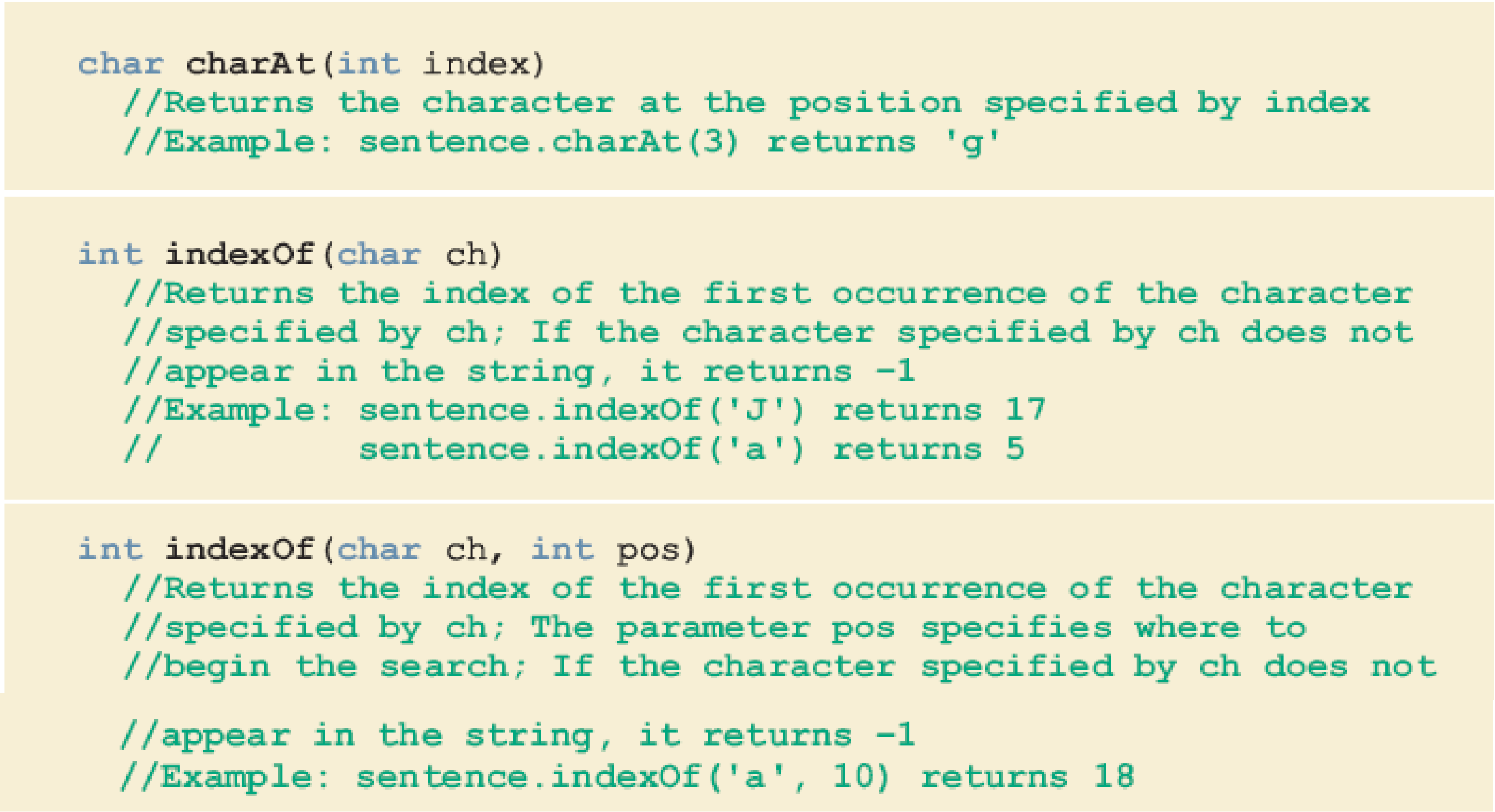
```
name = new String("Rashid Ali");
name = "Rashid Ali";
```

The class String (continued)

- A String object is an instance of class String
- The address of a String object with the value “Rashid Ali" is stored in name
- String methods can be used to modify string objects.

Some Commonly Used String Methods

TABLE 3-1 Some Commonly Used String Methods

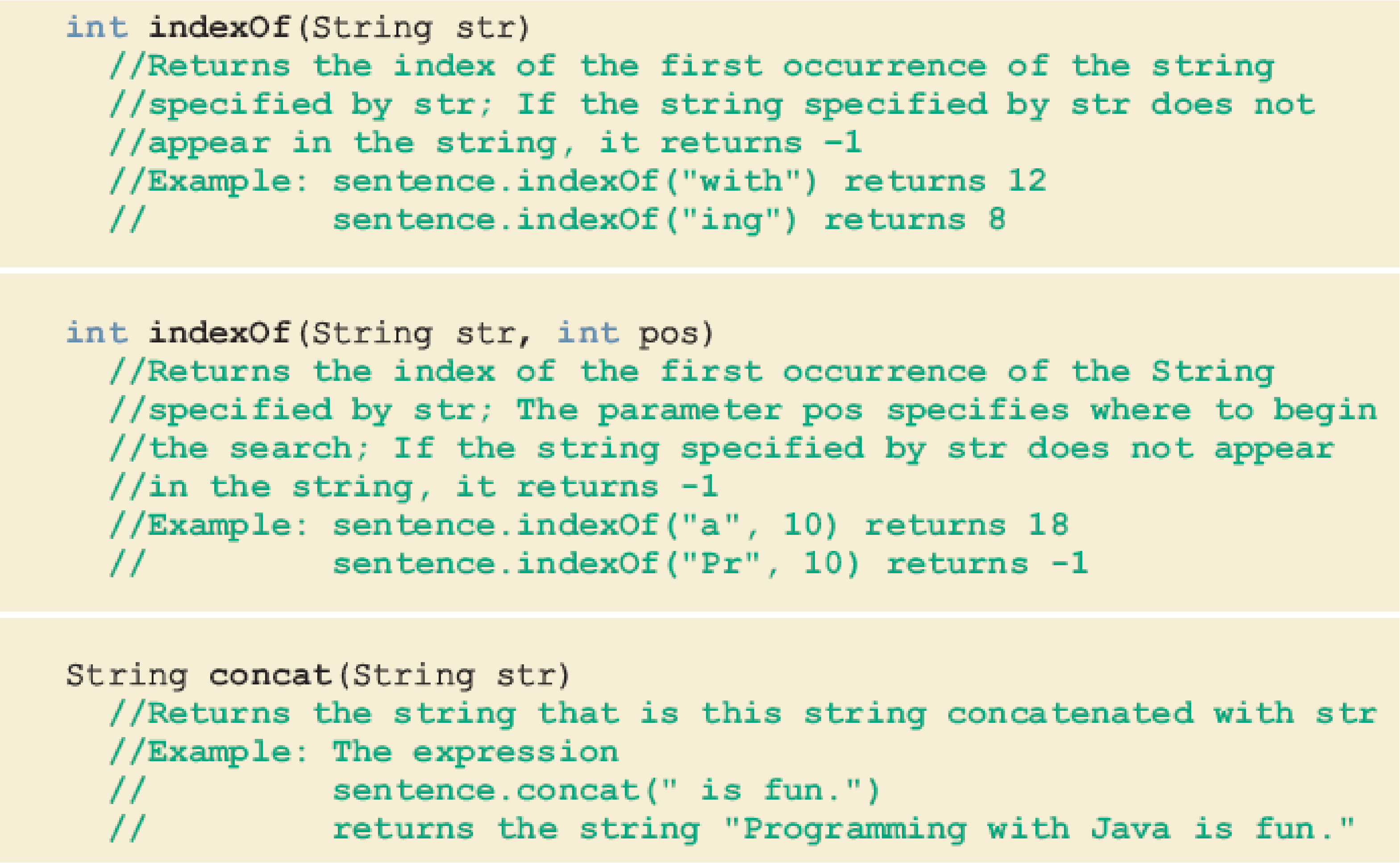


```
char charAt(int index)
  //Returns the character at the position specified by index
  //Example: sentence.charAt(3) returns 'g'

int indexOf(char ch)
  //Returns the index of the first occurrence of the character
  //specified by ch; If the character specified by ch does not
  //appear in the string, it returns -1
  //Example: sentence.indexOf('J') returns 17
  //         sentence.indexOf('a') returns 5

int indexOf(char ch, int pos)
  //Returns the index of the first occurrence of the character
  //specified by ch; The parameter pos specifies where to
  //begin the search; If the character specified by ch does not
  //appear in the string, it returns -1
  //Example: sentence.indexOf('a', 10) returns 18
```

Some Commonly Used String Methods (continued)

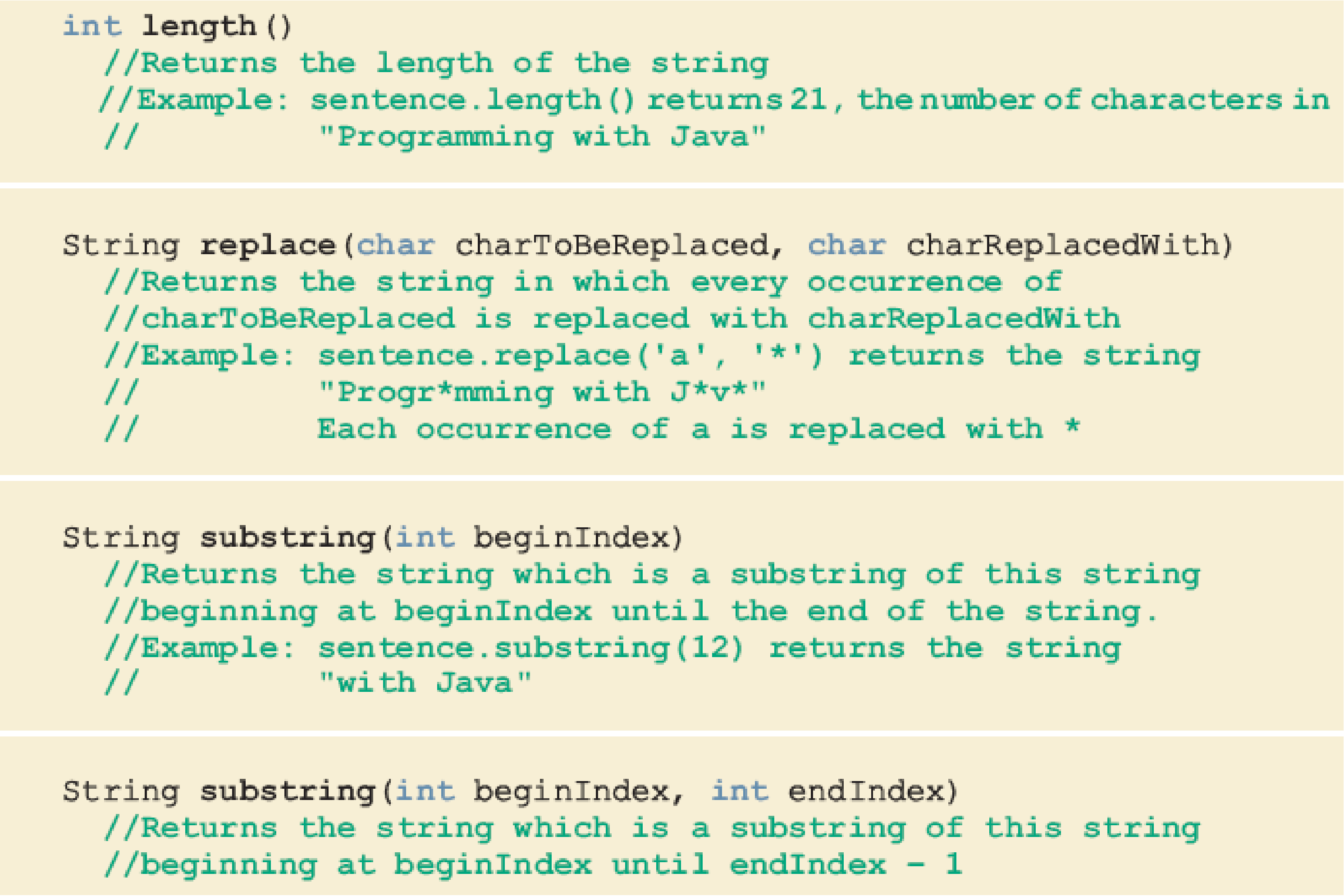


```
int indexOf(String str)
  //Returns the index of the first occurrence of the string
  //specified by str; If the string specified by str does not
  //appear in the string, it returns -1
  //Example: sentence.indexOf("with") returns 12
  //         sentence.indexOf("ing") returns 8
```

```
int indexOf(String str, int pos)
  //Returns the index of the first occurrence of the String
  //specified by str; The parameter pos specifies where to begin
  //the search; If the string specified by str does not appear
  //in the string, it returns -1
  //Example: sentence.indexOf("a", 10) returns 18
  //         sentence.indexOf("Pr", 10) returns -1
```

```
String concat(String str)
  //Returns the string that is this string concatenated with str
  //Example: The expression
  //         sentence.concat(" is fun.")
  //         returns the string "Programming with Java is fun."
```

Some Commonly Used String Methods (continued)



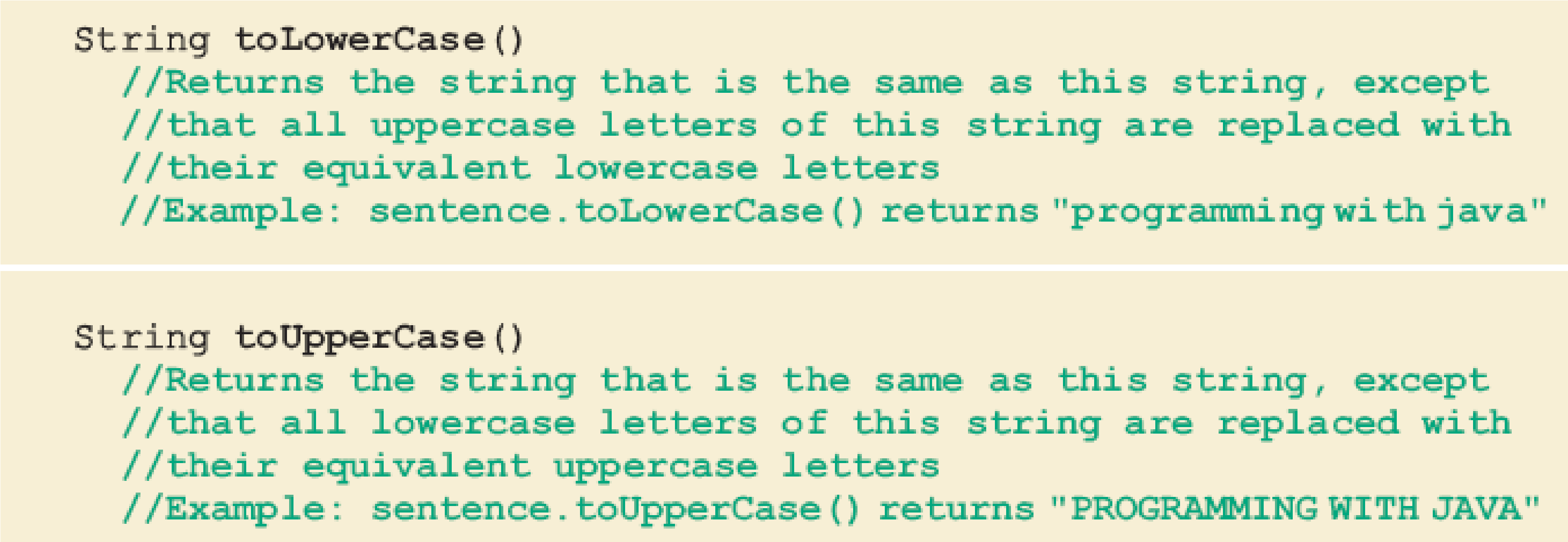
```
int length()
   //Returns the length of the string
   //Example: sentence.length() returns 21, the number of characters in
   //          "Programming with Java"

String replace(char charToBeReplaced, char charReplacedWith)
   //Returns the string in which every occurrence of
   //charToBeReplaced is replaced with charReplacedWith
   //Example: sentence.replace('a', '*') returns the string
   //          "Progr*mming with J*v*"
   //          Each occurrence of a is replaced with *

String substring(int beginIndex)
   //Returns the string which is a substring of this string
   //beginning at beginIndex until the end of the string.
   //Example: sentence.substring(12) returns the string
   //          "with Java"

String substring(int beginIndex, int endIndex)
   //Returns the string which is a substring of this string
   //beginning at beginIndex until endIndex - 1
```

Some Commonly Used String Methods (continued)



```
String toLowerCase()
  //Returns the string that is the same as this string, except
  //that all uppercase letters of this string are replaced with
  //their equivalent lowercase letters
  //Example: sentence.toLowerCase() returns "programming with java"
```

```
String toUpperCase()
  //Returns the string that is the same as this string, except
  //that all lowercase letters of this string are replaced with
  //their equivalent uppercase letters
  //Example: sentence.toUpperCase() returns "PROGRAMMING WITH JAVA"
```

Chapter Summary

- Primitive variables store data into their memory space
- Reference variables store the address of the object containing the data
- An object is an instance of a class

Chapter Summary (continued)

- Operator `new` is used to instantiate an object
- To use a predefined method you must know its name and the class and package it belongs to
- The dot (.) operator is used to access a certain method in a class

Chapter Summary (continued)

- Strings are objects in Java
- Methods of the `class String` are used to manipulate strings