Chapter 11

Section 11.2

11.2 Q1: Consider the examples below:
   A. a string.
   B. 'a string'.
   C. "a string".
   D. "1234".
   E. integer.

Which could be the value of a Java variable of type String?
   a. A and B.
   b. B and E.
   c. B and C.
   d. C and D.

ANS: d. C and D.

11.2 Q2: An anonymous String:
   a. has no value.
   b. is a string literal.
   c. can be changed.
   d. none of the above.

ANS: b. is a string literal.

Section 11.3

None

Section 11.3.1

11.3.1 Q1: A String constructor cannot be passed variables of type:
   a. char arrays.
   b. int arrays.
   c. byte arrays.
   d. Strings.

ANS: b. int arrays.

11.3.1 Q2: String objects are immutable. This means they:
   a. Must be initialized.
   b. Cannot be deleted.
   c. Cannot be changed.
   d. None of the above.

ANS: c. Cannot be changed.

Section 11.3.2

11.3.2 Q1: The length of a string can be determined by:
   a. The String method length().
   b. The String instance variable length.
   c. The String method strlen().
   d. All of the above.

ANS: a. The String method length().

11.3.2 Q2: How many String objects are instantiated by the following code segment (not including the literals)?

```
String s1, output;
s1 = "hello";
output += "\nThe string reversed is: "
for ( int i = s1.length() - 1; i >= 0; i-- )
```

© Copyright 2003 by Deitel & Associates, Inc. and Prentice Hall. All Rights Reserved.
Section 11.3.3

11.3.3 Q1: The statement
s1.equalsIgnoreCase( s4 )
is equivalent to which of the following?
a. s1.regionMatches( true, 0, s4, 0, s4.length() );
b. s1.regionMatches( 0, s4, 0, s4.length() );
c. s1.regionMatches( 0, s4, 0, s4.length() );
d. s1.regionMatches( true, s4, 0, s4.length() );
ANS: a. s1.regionMatches( true, 0, s4, 0, s4.length() );

11.3.3 Q2: The statement
s1.startsWith( "art" )
has the same result as which of the following?
a. s1.regionMatches( 0, "art", 0, 3 );
b. s2 = s1.getChars( 0, 3 );
   s2.equals( "art" );
c. s1.regionMatches( true, 0, "art", 0, 3 );
d. All of the above
ANS: d. All of the above

Section 11.3.4

11.3.4 Q1: For
String c = "hello world";
The Java statements
   int i = c.indexOf( 'o' );
   int j = c.lastIndexOf( 'l' );
will result in:
a. i = 4 and j = 8.
b. i = 5 and j = 8.
c. i = 4 and j = 9.
d. i = 5 and j = 9.
ANS: c. i = 4 and j = 9.

11.3.4 Q2: For
String c = "Hello. She sells sea shells";
The Java statements
   int i = c.indexOf( "ll" );
   int j = c.lastIndexOf( "ll" );
will result in:
a. i = 2 and j = 24.
b. i = 3 and j = 24.
c. i = 2 and j = 25.
d. i = 3 and j = 23.
ANS: a. i = 2 and j = 24.

Section 11.3.5

11.3.5 Q1: For
String c = "Now is the time for all";
The Java statements
String i = c.substring(7);
String j = c.substring(4, 15);

will result in:

a. i = "he time for all" and j = "is the time".

b. i = "the time for all" and j = "s the time".

c. i = "the time for all" and j = "is the time ".

d. i = "he time for all" and j = "s the time".

ANS: c. i = "the time for all" and j = "is the time ".

11.3.5 Q2: The String method substring returns:

a. A char.
b. A String.
c. void.
d. A char[].


Section 11.3.6

11.3.6 Q1: Consider the statements below:

String a = "JAVA: ";
String b = "How to ";
String c = "Program";

Which of the statements below will create the String r1 = "JAVA: How to Program"?

a. String r1 = c.concat( b.concat( a ) );
b. String r1 = a.concat( b.concat( c ) );
c. String r1 = b.concat( c.concat( a ) );
d. String r1 = c.concat( c.concat( b ) );

ANS: b. String r1 = a.concat( b.concat( c ) );

Section 11.3.7

11.3.7 Q1: Consider the String below:

String r = "a toyota";

Which of the following will create the String r1 = "a TOYOTa"?

a. r1 = r.replace( "toyot", TOYOT" );
b. r1 = r.replace( 't','T' );
   r1 = r.replace( 'o','0' );
   r1 = r.replace( 'y','Y' );
c. r1 = r.replace( 't','T' ).replace( 'o', '0' ).replace( 'y', 'Y' );
d. r1 = r.substring( 2, 4 ).toUpperCase();

ANS: c. r1 = r.replace( 't','T' ).replace( 'o', '0' ).replace( 'y', 'Y' );

11.3.7 Q2: Which of the following is not a method of class String?

a. toUpperCase.
b. trim.
c. toCharacterArray.
d. All of the above are methods of class String.

ANS: c. toCharacterArray.

Section 11.3.8

11.3.8 Q1: Which of the following will create a String different from the other three?

a. String r = "123456"
b. int i = 123;
   int j = 456;
   String r = String.valueOf(j) + String.valueOf(i);
c. int i = 123;
int j = 456;
String r = String.valueOf(i) + String.valueOf(j);
d.
int i = 123;
int j = 456;
String r = i + j;
ANS: b. int i = 123;
int j = 456;
String r = String.valueOf(j) + String.valueOf(i);

Section 11.4

11.13 Q1: StringBuffer objects can be used in place of String objects if:
   a. The string data is not constant.
   b. The string data size may grow.
   c. Performance is not critical.
   d. All of the above.
   ANS: d. All of the above.

Section 11.4.1

11.4.1 Q1: Given the following declarations:
   StringBuffer buf;
   StringBuffer buf2 = new StringBuffer();
   String c = new String( "test" );
Which of the following is not a valid StringBuffer constructor?
   a. buf = new StringBuffer();
   b. buf = new StringBuffer( buf2 );
   c. buf = new StringBuffer( 32 );
   d. buf = new StringBuffer( c );
   ANS: b. buf = new StringBuffer( buf2 );

11.4.1 Q2: Given the following declaration:
   StringBuffer buf = new StringBuffer();
   What is the capacity of buf?
   a. 0
   b. 10
   c. 16
   d. 20
   ANS: c. 16

Section 11.4.2

11.4.2 Q1: Which of the following statements is true?
   a. The capacity of a StringBuffer is equal to its length.
   b. The capacity of a StringBuffer cannot exceed its length.
   c. The length of a StringBuffer cannot exceed its capacity.
   d. Both a and b are true.
   ANS: c. The length of a StringBuffer cannot exceed its capacity.

11.4.2 Q2: Given the following declarations:
   StringBuffer buffer = new StringBuffer( "Testing Testing" );
   buffer.setLength( 7 );
   buffer.ensureCapacity( 5 );
Which of the following is true?
   a. buffer has capacity 5.
   b. buffer has capacity 31.
   c. buffer has content “Testin”.
   d. buffer has length 15.
ANS: b. buffer has capacity 31.

Section 11.4.3

11.4.3 Q1: Consider the statement below:

```java
StringBuffer sb1 = new StringBuffer( "a toyota" );
```

Which of the following creates a `String` object with the value "toy"?

a. `String res = sb1.substring( 2, 5 );`
   - sb1.getChars( 2, 5, dest, 0 );
   - String res = new String(dest);

b. `char dest[] = new char[ sb1.length() ];`
   - dest = sb1.getChars( 2, 5 );
   - String res = new String( dest );

b. `char dest[] = new char[ sb1.length() ];`
   - dest = sb1.getChars( 2, 5 );
   - String res = new String( dest );

c. `char dest[] = new char[ sb1.length() ];`
   - dest = sb1.getChars( 0, 3 );
   - String res = new String( dest );

ANS: b. `char dest[] = new char[ sb1.length() ];`
   - dest = sb1.getChars( 2, 5 );
   - String res = new String( dest );

11.4.3 Q2: To find the character at a certain index position within a `String`, use the method:

a. `getChars`, with the index as an argument.

b. `getCharAt`, with the index as an argument.

c. `charAt`, with the index as an argument.

ANS: c. `charAt`, with the index as an argument.

Section 11.4.4

11.4.4 Q1: Which of the following creates the string of the numbers from 1 to 1000 most efficiently?

a. `String s;`
   - for ( int i = 1; i <= 1000; i++ )
     - s += i;

b. `StringBuffer sb = new StringBuffer( 10 );`
   - for ( int i = 1; i <= 1000; i++ )
     - sb.append( i );
   - String s = new String( sb );

c. `StringBuffer sb = new StringBuffer( 3000 );`
   - for ( int i = 1; i <= 1000; i++ )
     - sb.append( i );
   - String s = new String( sb );

d. All are equivalently efficient.

ANS: c. `StringBuffer sb = new StringBuffer( 3000 );`
   - for ( int i = 1; i <= 1000; i++ )
     - sb.append( i );
   - String s = new String( sb );

Section 11.4.5

11.4.5 Q1: Consider the statements below:

```java
StringBuffer sb = new StringBuffer( "a toyota" );
sb.insert( 2, "landrover" );
sb.delete( 11, 16 );
sb.insert( 11, " " );
```

The `StringBuffer` contents at the end of this segment will be:
Section 11.5

11.5 Q1: Which of the following are static Character methods?
   a. Character.hashcode( char c );
   b. Character.isDigit( char c );
   c. Character.equals( char c );
   d. All of the above.

ANS: b. Character.isDigit( char c );

11.5 Q2: Which class is not a type-wrapper class?
   a. Character
   b. Int
   c. Double
   d. Byte

ANS: b. Int

Section 11.6

11.6 Q1: Consider the Java segment:
   String line1 = new String( "c = 1 + 2 + 3" ) ;
   StringTokenizer tok = new StringTokenizer( line1, delimArg );

For the String line1 to have 4 tokens, delimArg should be:
   a. String delimArg = "+=";
   b. String delimArg = "123"
   c. String delimArg = "c+";
   d. String delimArg = " ";

ANS: a. String delimArg = "+=";

11.6 Q2: Consider the Java segment:
   String line1 = new String( "c = 1 + 2 + 3" ) ;
   StringTokenizer tok = new StringTokenizer( line1, delimArg );
   int count = tok.countTokens();

The value of count is:
   a. 8.
   b. 7.
   c. 13.
   d. 4.

ANS: b. 7.

11.6 Q3: Consider the Java segment:
   String line1 = new String( "c = 1 + 2 + 3" ) ;
   StringTokenizer tok = new StringTokenizer( line1, "+=" );

   String foo = tok.nextToken();
   String bar = tok.nextToken();

The values of foo and bar are:
   a. foo is "c", bar is " = ".
   b. foo is "c", bar is " ".
   c. foo is " = ", bar is " + ".
   d. foo is "c", bar is " 1 ".

© Copyright 2003 by Deitel & Associates, Inc. and Prentice Hall. All Rights Reserved.
ANS: d. **foo** is “c ”, **bar** is “ l ”.

Section 11.7

None

Section 11.8

11.8: Q1: Which of the following is not a word character?
   a. w
   b. 0
   c. _
   d. &
   ANS: d. &.

11.8 Q2: Which of the following statements is true?
   a. Ranges of characters can be represented by placing a ~ between two characters.
   b. [^Z] is the same as [A−Y].
   c. Both “A*” and “A+” will match “A”, but only “A+” will match an empty string.
   d. All of above.
   ANS: c. Both “A*” and “A+” will match “A”, but only “A+” will match an empty string.

**Replacing Substrings and Splitting Strings**

11.8: Q3: Which of the Java strings represent the regular expression,\s*?
   a. “\s\*”.
   b. “,\s*”.
   c. “,\s\*”.
   d. “.\s\*”.
   ANS: b. “,\s\*”.

**Classes Pattern and Matcher**

11.8: Q4: Which of the following statements is true?
   a. Class Matcher provides methods find, lookingAt, replaceFirst and replaceAll.
   b. Method matches (from class String, Pattern or Matcher) will return true only if the entire search object matches the regular expression.
   c. Methods find and lookingAt (from class Matcher) will return true if a portion of the search object matches the regular expression.
   d. All of above.
   ANS: d. All of above.