**Chapter 9 Strings**

*Section 9.2 The String Class*

***9.1***  Suppose s is a string with the value "java". What will be assigned to x if you execute the following code?  
  
char x = s.charAt(4);

 A. 'a'

 B. 'v'

 C. Nothing will be assigned to x, because the execution causes the runtime error StringIndexOutofBoundsException.

Your answer is correct http://www.cs.armstrong.edu/liang/image/correct.jpg

***9.2***  Which of the following statements is preferred to create a string "Welcome to Java"?

 A. String s = "Welcome to Java";

 B. String s = new String("Welcome to Java");

 C. String s; s = "Welcome to Java";

 D. String s; s = new String("Welcome to Java");

Your answer B is incorrect http://www.cs.armstrong.edu/liang/image/wrong.jpg

The correct answer is A  
Explanation: (a) is better than (b) because the string created in (a) is interned. Since strings are immutable and are ubiquitous in programming, to improve efficiency and save memory, the JVM uses a unique instance for string literals with the same character sequence. Such an instance is called interned. The JVM (a) is simpler than (c).

***9.3***  What is the output of the following code?  
  
public class Test {    
  public static void main(String[] args) {  
    String s1 = "Welcome to Java!";  
    String s2 = s1;  
  
    if (s1 == s2)  
      System.out.println("s1 and s2 reference to the same String object");  
    else  
      System.out.println("s1 and s2 reference to different String objects");  
  }  
}

 A. s1 and s2 reference to the same String object

 B. s1 and s2 reference to different String objects

Your answer B is incorrect http://www.cs.armstrong.edu/liang/image/wrong.jpg

The correct answer is A

***9.4***  What is the output of the following code?  
  
public class Test {    
  public static void main(String[] args) {  
    String s1 = "Welcome to Java!";  
    String s2 = "Welcome to Java!";  
  
    if (s1 == s2)  
      System.out.println("s1 and s2 reference to the same String object");  
    else  
      System.out.println("s1 and s2 reference to different String objects");  
  }  
}

 A. s1 and s2 reference to the same String object

 B. s1 and s2 reference to different String objects

Your answer B is incorrect http://www.cs.armstrong.edu/liang/image/wrong.jpg

The correct answer is A  
Explanation: Since strings are immutable and are ubiquitous in programming, to improve efficiency and save memory, the JVM uses a unique instance for string literals with the same character sequence. Such an instance is called interned.

***9.5***  What is the output of the following code?  
  
public class Test {    
  public static void main(String[] args) {  
    String s1 = new String("Welcome to Java!");  
    String s2 = new String("Welcome to Java!");  
  
    if (s1 == s2)  
      System.out.println("s1 and s2 reference to the same String object");  
    else  
      System.out.println("s1 and s2 reference to different String objects");  
  }  
}

 A. s1 and s2 reference to the same String object

 B. s1 and s2 reference to different String objects

Your answer A is incorrect http://www.cs.armstrong.edu/liang/image/wrong.jpg

The correct answer is B

***9.6***  Suppose s1 and s2 are two strings. What is the result of the following code?  
  
    s1.equals(s2) == s2.equals(s1)

 A. true

 B. false

Your answer B is incorrect http://www.cs.armstrong.edu/liang/image/wrong.jpg

The correct answer is A

***9.7***  What is the output of the following code?  
  
public class Test {   
  public static void main(String[] args) {  
    String s1 = new String("Welcome to Java!");  
    String s2 = new String("Welcome to Java!");  
  
    if (s1.equals(s2))  
      System.out.println("s1 and s2 have the same contents");  
    else  
      System.out.println("s1 and s2 have different contents");  
  }  
}

 A. s1 and s2 have the same contents

 B. s1 and s2 have different contents

Your answer is correct http://www.cs.armstrong.edu/liang/image/correct.jpg

***9.8***  What is the output of the following code?  
  
public class Test {    
  public static void main(String[] args) {  
    String s1 = new String("Welcome to Java!");  
    String s2 = s1.toUpperCase();  
  
    if (s1 == s2)  
      System.out.println("s1 and s2 reference to the same String object");  
    else if (s1.equals(s2))  
      System.out.println("s1 and s2 have the same contents");  
    else  
      System.out.println("s1 and s2 have different contents");  
  }  
}

 A. s1 and s2 reference to the same String object

 B. s1 and s2 have the same contents

 C. s1 and s2 have different contents

Your answer is correct http://www.cs.armstrong.edu/liang/image/correct.jpg

***9.9***  What is the output of the following code?  
  
public class Test {    
  public static void main(String[] args) {  
    String s1 = new String("Welcome to Java");  
    String s2 = s1;  
  
    s1 += "and Welcome to HTML";  
  
    if (s1 == s2)  
      System.out.println("s1 and s2 reference to the same String object");  
    else  
      System.out.println("s1 and s2 reference to different String objects");  
  }  
}

 A. s1 and s2 reference to the same String object

 B. s1 and s2 reference to different String objects

Your answer is correct http://www.cs.armstrong.edu/liang/image/correct.jpg

***9.10***  Suppose s1 and s2 are two strings. Which of the following statements or expressions are incorrect?

 A. String s = new String("new string");

 B. String s3 = s1 + s2

 C. s1 >= s2

 D. int i = s1.length

 E. s1.charAt(0) = '5'

Your answer is correct http://www.cs.armstrong.edu/liang/image/correct.jpg

***9.11***  Suppose s1 and s2 are two strings. Which of the following statements or expressions is incorrect?

 A. String s3 = s1 - s2;

 B. boolean b = s1.compareTo(s2);

 C. char c = s1[0];

 D. char c = s1.charAt(s1.length());

Your answer is correct http://www.cs.armstrong.edu/liang/image/correct.jpg

***9.12***  "abc".compareTo("aba") returns \_\_\_\_\_\_\_\_\_\_\_.

 A. 1

 B. 2

 C. -1

 D. -2

 E. 0

Your answer is correct http://www.cs.armstrong.edu/liang/image/correct.jpg

***9.13***  "AbA".compareToIgnoreCase("abC") returns \_\_\_\_\_\_\_\_\_\_\_.

 A. 1

 B. 2

 C. -1

 D. -2

 E. 0

Your answer is correct http://www.cs.armstrong.edu/liang/image/correct.jpg

***9.14***  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ returns true.

 A. "peter".compareToIgnoreCase("Peter")

 B. "peter".compareToIgnoreCase("peter")

 C. "peter".equalsIgnoreCase("Peter")

 D. "peter".equalsIgnoreCase("peter")

 E. "peter".equals("peter")

Your answer is correct http://www.cs.armstrong.edu/liang/image/correct.jpg

***9.15***  What is the output of the following code?  
  
String s = "University";  
s.replace("i", "ABC");  
System.out.println(s);

 A. UnABCversity

 B. UnABCversABCty

 C. UniversABCty

 D. University

Your answer is correct http://www.cs.armstrong.edu/liang/image/correct.jpg

***9.16***  What is the return value of "SELECT".substring(0, 5)?

 A. "SELECT"

 B. "SELEC"

 C. "SELE"

 D. "ELECT"

Your answer is correct http://www.cs.armstrong.edu/liang/image/correct.jpg

***9.17***  What is the return value of "SELECT".substring(4, 4)?

 A. an empty string

 B. C

 C. T

 D. E

Your answer is correct http://www.cs.armstrong.edu/liang/image/correct.jpg

***9.18***  Analyze the following code.  
  
class Test {    
  public static void main(String[] args) {  
    String s;  
    System.out.println("s is " + s);  
  }  
}

 A. The program has a compilation error because s is not initialized, but it is referenced in the println statement.

 B. The program has a runtime error because s is not initialized, but it is referenced in the println statement.

 C. The program has a runtime error because s is null in the println statement.

 D. The program compiles and runs fine.

Your answer is correct http://www.cs.armstrong.edu/liang/image/correct.jpg

***9.19***  To check if a string s contains the prefix "Java", you may write 

 A. if (s.startsWith("Java")) ...

 B. if (s.indexOf("Java") == 0) ...

 C. if (s.substring(0, 4).equals("Java")) ...

 D. if (s.charAt(0) == 'J' && s.charAt(1) == 'a' && s.charAt(2) == 'v' && s.charAt(3) == 'a') ...

Your answer is correct http://www.cs.armstrong.edu/liang/image/correct.jpg

***9.20***  To check if a string s contains the suffix "Java", you may write 

 A. if (s.endsWith("Java")) ...

 B. if (s.lastIndexOf("Java") >= 0) ...

 C. if (s.substring(s.length() - 4).equals("Java")) ...

 D. if (s.substring(s.length() - 5).equals("Java")) ...

 E. if (s.charAt(s.length() - 4) == 'J' && s.charAt(s.length() - 3) == 'a' && s.charAt(s.length() - 2) == 'v' && s.charAt(s.length() - 1) == 'a') ...

Your answer is correct http://www.cs.armstrong.edu/liang/image/correct.jpg

***9.21***  Which of the following is the correct statement to return JAVA?

 A. toUpperCase("Java")

 B. "Java".toUpperCase("Java")

 C. "Java".toUpperCase()

 D. String.toUpperCase("Java")

Your answer A is incorrect http://www.cs.armstrong.edu/liang/image/wrong.jpg

The correct answer is C

***9.22***  Which of the following is the correct statement to return a string from an array a of characters?

 A. toString(a)

 B. new String(a)

 C. convertToString(a)

 D. String.toString(a)

Your answer is correct http://www.cs.armstrong.edu/liang/image/correct.jpg

***9.23***  Assume s is " abc ", the method \_\_\_\_\_\_\_\_\_\_ returns a new string "abc".

 A. s.trim(s)

 B. trim(s)

 C. String.trim(s)

 D. s.trim()

Your answer is correct http://www.cs.armstrong.edu/liang/image/correct.jpg

***9.24***  Assume s is "ABCABC", the method \_\_\_\_\_\_\_\_\_\_ returns a new string "aBCaBC".

 A. s.toLowerCase(s)

 B. s.toLowerCase()

 C. s.replace('A', 'a')

 D. s.replace('a', 'A')

 E. s.replace("ABCABC", "aBCaBC")

Your answer C is incorrect http://www.cs.armstrong.edu/liang/image/wrong.jpg

The correct answer is CE

***9.25***  Assume s is "ABCABC", the method \_\_\_\_\_\_\_\_\_\_ returns an array of characters.

 A. toChars(s)

 B. s.toCharArray()

 C. String.toChars()

 D. String.toCharArray()

 E. s.toChars()

Your answer is correct http://www.cs.armstrong.edu/liang/image/correct.jpg

***9.26***  \_\_\_\_\_\_\_\_\_\_ returns a string.

 A. String.valueOf(123)

 B. String.valueOf(12.53)

 C. String.valueOf(false)

 D. String.valueOf(new char[]{'a', 'b', 'c'})

Your answer is correct http://www.cs.armstrong.edu/liang/image/correct.jpg

***9.27***  The following program displays \_\_\_\_\_\_\_\_\_\_.  
  
public class Test {    
  public static void main(String[] args) {  
    String s = "Java";  
    StringBuilder buffer = new StringBuilder(s);  
    change(s);  
    System.out.println(s);  
  }  
    
  private static void change(String s) {  
    s = s + " and HTML";  
  }  
}

 A. Java

 B. Java and HTML

 C. and HTML

 D. nothing is displayed

Your answer B is incorrect http://www.cs.armstrong.edu/liang/image/wrong.jpg

The correct answer is A  
Explanation: Inside the method, the statement s = s + ' and HTML' creates a new String object s, which is different from the original String object passed to the change(s) method. The original String object has not been changed. Therefore, the output from the original string is Java.

***9.28***  What is displayed by the following statement?  
        System.out.println("Java is neat".replaceAll("is", "AAA"));

 A. JavaAAAneat

 B. JavaAAA neat

 C. Java AAA neat

 D. Java AAAneat

Your answer is correct http://www.cs.armstrong.edu/liang/image/correct.jpg

***9.29***  What is displayed by the following code?  
  public static void main(String[] args) {  
    String[] tokens = "Welcome to Java".split("o");  
    for (int i = 0; i < tokens.length; i++) {  
      System.out.print(tokens[i] + " ");  
    }  
  }

 A. Welcome to Java

 B. Welc me to Java

 C. Welc me t Java

 D. Welcome t Java

Your answer is correct http://www.cs.armstrong.edu/liang/image/correct.jpg

***9.30***  What is displayed by the following code?  
    System.out.print("Hi, ABC, good".matches("ABC ") + " ");  
    System.out.println("Hi, ABC, good".matches(".\*ABC.\*"));

 A. false fasle

 B. true fasle

 C. true true

 D. false true

Your answer is correct http://www.cs.armstrong.edu/liang/image/correct.jpg

***9.31***  What is displayed by the following code?  
    System.out.print("A,B;C".replaceAll(",;", "#") + " ");  
    System.out.println("A,B;C".replaceAll("[,;]", "#"));

 A. A B C A#B#C

 B. A#B#C A#B#C

 C. A,B;C A#B#C

 D. A B C A B C

Your answer is correct http://www.cs.armstrong.edu/liang/image/correct.jpg

***9.32***  What is displayed by the following code?  
  
    String[] tokens = "A,B;C;D".split("[,;]");  
    for (int i = 0; i < tokens.length; i++)  
      System.out.print(tokens[i] + " ");

 A. A,B;C;D

 B. A B C D

 C. A B C;D

 D. A B;C;D

Your answer is correct http://www.cs.armstrong.edu/liang/image/correct.jpg

*Section 9.5 The Character Class*

***9.33***  Which of following is not a correct method in Character?

 A. isLetterOrDigit(char)

 B. isLetter(char)

 C. isDigit()

 D. toLowerCase(char)

 E. toUpperCase()

Your answer is correct http://www.cs.armstrong.edu/liang/image/correct.jpg

***9.34***  Suppose Character x = new Character('a'), \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ returns true.

 A. x.equals(new Character('a'))

 B. x.compareToIgnoreCase('A')

 C. x.equalsIgnoreCase('A')

 D. x.equals('a')

 E. x.equals("a")

Your answer is correct http://www.cs.armstrong.edu/liang/image/correct.jpg

*Section 9.6 The StringBuilder/StringBuffer Class*

***9.35***  Analyze the following code.  
  
class Test {    
  public static void main(String[] args) {  
    StringBuilder strBuf = new StringBuilder(4);  
    strBuf.append("ABCDE");  
    System.out.println("What's strBuf.charAt(5)? " + strBuf.charAt(5));  
  }  
}

 A. The program has a compilation error because you cannot specify initial capacity in the StringBuilder constructor.

 B. The program has a runtime error because because the buffer's capacity is 4, but five characters "ABCDE" are appended into the buffer.

 C. The program has a runtime error because the length of the string in the buffer is 5 after "ABCDE" is appended into the buffer. Therefore, strBuf.charAt(5) is out of range.

 D. The program compiles and runs fine.

Your answer A is incorrect http://www.cs.armstrong.edu/liang/image/wrong.jpg

The correct answer is C  
Explanation: The charAt method returns the character at a specific index in the string buffer. The first character of a string buffer is at index 0, the next at index 1, and so on. The index argument must be greater than or equal to 0, and less than the length of the string buffer.

***9.36***  Which of the following is true?

 A. You can add characters into a string buffer.

 B. You can delete characters into a string buffer.

 C. You can reverse the characters in a string buffer.

 D. The capacity of a string buffer can be automatically adjusted.

Your answer is correct http://www.cs.armstrong.edu/liang/image/correct.jpg

***9.37***  \_\_\_\_\_\_\_\_\_ returns the last character in a StringBuilder variable named strBuf?

 A. strBuf.charAt(strBuf.length() - 1)

 B. strBuf.charAt(strBuf.capacity() - 1)

 C. StringBuilder.charAt(strBuf.length() - 1)

 D. StringBuilder.charAt(strBuf.capacity() - 1)

Your answer is correct http://www.cs.armstrong.edu/liang/image/correct.jpg

***9.38***  Assume StringBuilder strBuf is "ABCDEFG", after invoking \_\_\_\_\_\_\_\_\_, strBuf contains "AEFG".

 A. strBuf.delete(0, 3)

 B. strBuf.delete(1, 3)

 C. strBuf.delete(1, 4)

 D. strBuf.delete(2, 4)

Your answer is correct http://www.cs.armstrong.edu/liang/image/correct.jpg

***9.39***  Assume StringBuilder strBuf is "ABCDEFG", after invoking \_\_\_\_\_\_\_\_\_, strBuf contains "ABCRRRRDEFG".

 A. strBuf.insert(1, "RRRR")

 B. strBuf.insert(2, "RRRR")

 C. strBuf.insert(3, "RRRR")

 D. strBuf.insert(4, "RRRR")

Your answer is correct http://www.cs.armstrong.edu/liang/image/correct.jpg

***9.40***  Assume StringBuilder strBuf is "ABCCEFC", after invoking \_\_\_\_\_\_\_\_\_, strBuf contains "ABTTEFT".

 A. strBuf.replace('C', 'T')

 B. strBuf.replace("C", "T")

 C. strBuf.replace("CC", "TT")

 D. strBuf.replace('C', "TT")

 E. strBuf.replace(2, 7, "TTEFT")

Your answer B is incorrect http://www.cs.armstrong.edu/liang/image/wrong.jpg

The correct answer is E

***9.41***  The StringBuilder methods \_\_\_\_\_\_\_\_\_\_\_\_\_ not only change the contents of a string buffer, but also returns a reference to the string buffer.

 A. delete

 B. append

 C. insert

 D. reverse

 E. replace

Your answer is correct http://www.cs.armstrong.edu/liang/image/correct.jpg

***9.42***  The following program displays \_\_\_\_\_\_\_\_\_\_.  
  
public class Test {    
  public static void main(String[] args) {  
    String s = "Java";  
    StringBuilder buffer = new StringBuilder(s);  
    change(buffer);  
    System.out.println(buffer);  
  }  
    
  private static void change(StringBuilder buffer) {  
    buffer.append(" and HTML");  
  }  
}

 A. Java

 B. Java and HTML

 C. and HTML

 D. nothing is displayed

Your answer is correct http://www.cs.armstrong.edu/liang/image/correct.jpg

*Section 9.7 Command-Line Arguments*

***9.43***  How can you get the word "abc" in the main method from the following call?  
  
java Test "+" 3 "abc" 2

 A. args[0]

 B. args[1]

 C. args[2]

 D. args[3]

Your answer D is incorrect http://www.cs.armstrong.edu/liang/image/wrong.jpg

The correct answer is C

***9.44***  Given the following program:  
  
public class Test {  
  public static void main(String[] args) {  
    for (int i = 0; i < args.length; i++) {  
      System.out.print(args[i] + " ");  
    }  
  }  
}  
  
What is the output, if you run the program using   
  
java Test 1 2 3

 A. 3

 B. 1

 C. 1 2 3

 D. 1 2

Your answer is correct http://www.cs.armstrong.edu/liang/image/correct.jpg

***9.45***  Which code fragment would correctly identify the number of arguments passed via the command line to a Java application, excluding the name of the class that is being invoked?

 A. int count = args.length;

 B. int count = args.length - 1;

 C. int count = 0; while (args[count] != null) count ++;

 D. int count=0; while (!(args[count].equals(""))) count ++;

Your answer D is incorrect http://www.cs.armstrong.edu/liang/image/wrong.jpg

The correct answer is A

***9.46***  Which correctly creates an array of five empty Strings?

 A. String[] a = new String [5];

 B. String[] a = {"", "", "", "", ""};

 C. String[5] a;

 D. String[ ] a = new String [5]; for (int i = 0; i < 5; a[i++] = null);

Your answer A is incorrect http://www.cs.armstrong.edu/liang/image/wrong.jpg

The correct answer is B

***9.47***  Identify the problems in the following code.  
                  
public class Test {  
  public static void main(String argv[]) {  
    System.out.println("argv.length is " + argv.length);  
  }  
}

 A. The program has a compile error because String argv[] is wrong and it should be replaced by String[] args.

 B. The program has a compile error because String args[] is wrong and it should be replaced by String args[].

 C. If you run this program without passing any arguments, the program would have a runtime error because argv is null.

 D. If you run this program without passing any arguments, the program would display argv.length is 0.

Your answer A is incorrect http://www.cs.armstrong.edu/liang/image/wrong.jpg

The correct answer is D  
Explanation: The parameter for the main method is an array of String. The declaration String argv[] is correct. When you run the program without passing arguments, argv is new String[0]. Thus, argv.length is 0. See the NOTE box in the section, 'Passing Arguments to Java Programs.'

***9.48***  Which of the following is the correct header of the main method?

 A. public static void main(String[] args)

 B. public static void main(String args[])

 C. public static void main(String[] x)

 D. public static void main(String x[])

 E. static void main(String[] args)

Your answer B is incorrect http://www.cs.armstrong.edu/liang/image/wrong.jpg

The correct answer is ABCD  
Explanation: e is incorrect because the main method must be public.