

Solutions to Quick Check Questions

8

Characters and Strings

9.1 Characters

1. Determine the output of the following statements:

a. `System.out.println((char) 65);`

A

b. `System.out.println((int) 'C');`

67

c. `System.out.println('Y');`

Y

d. `if ('A' < '?')
 System.out.println('A');
else
 System.out.println('?');`

?

2. How many distinct characters can you represent by using eight bits?

$$2^8 = 256$$

9.2 Strings

1. Determine the output of the following code:

```
a.   String str = "Programming";
      for (int i = 0; i < 9; i+=2) {
          System.out.print( str.charAt( i ) );
      }
```

Pormi

```
b.   String str = "World Wide Web";
      for (int i = 0; i < 10; i ++ ) {
          if ( str.charAt(i) == 'W') {
              System.out.println( 'M' );
          }
          else {
              System.out.print( str.charAt(i) );
          }
      }
```

*M
orld
M
ide*

2. Write a loop that prints out a string in reverse. If the string is Hello then the code outputs olleH. Use System.out.

```
int max = str.length()-1;
for (int i = max; i >= 0; i--)
    System.out.print( str.charAt(i) );
}
```

3. Assuming two String objects str1 and str2 are initialized as follows:

```
String str1 = "programming";
String str2 = "language";
```

Determine the value of each of the following expressions if they are valid. If they are not valid, state the reason why.

a. `str1.compareTo(str2)`

positive number

b. `str2.compareTo(str2)`

0

c. `str2.substring(1, 1)`

"" //empty string

d. `str2.substring(0, 7);`

"languag"

e. `str2.charAt(11);`

invalid —out of bounds error

f. `str1.length() + str2.length()`

19

4. What is the difference between the two String methods `equals` and `equalsIgnoreCase`?

equals is a case-sensitive comparison while equalsIgnoreCase is not.

9.3 Pattern Matching and Regular Expression

1. Describe the string the following regular expression matches:

a. `a*b`

b. `b[aiu]d`

c. `[Oo]bject(s|)`

a. *Strings that begins with 'a' and terminates with 'b'*

b. *Strings "bad", "bid", and "bud"*

c. *Strings "Objects", "objects", "Object ", and "object "*

2. Write a regular expression for a state vehicle license number whose format is a single capital letter, followed by three digits and four lowercase letters.

`[A-Z][0-9]{3}[a-z]{4}`

3. Which of the following regular expressions are invalid?

a. `(a-z)*+`

b. `[a|ab]xyz`

c. `abe-14`

d. `[a-z&&^a^b]`

e. `[//one]two`

They are all syntactically valid, but from a logical standpoint, expressions (a) and (e) are not correct.

9.4 The Pattern and Matcher Classes

1. Replace the following statements with the equivalent ones using the Pattern and Matcher classes:

a. `str.replaceAll("1", "one");`

b. `str.matches("alpha");`

Answers:

a.

```
Pattern pattern = Pattern.compile("1");
Matcher matcher = pattern.matcher(str);
```

```
matcher.replaceAll("one");
```

b.

```
Pattern pattern = Pattern.compile("alpha");
Matcher matcher = pattern.matcher(str);
matcher.matches();
```

2. Using the `find` method of the `Matcher` class, check if the given string document contains the whole word `Java`.

```
Pattern pattern = Pattern.compile("Java",
    Pattern.CASE_SENSITIVE);

String document = ... ;

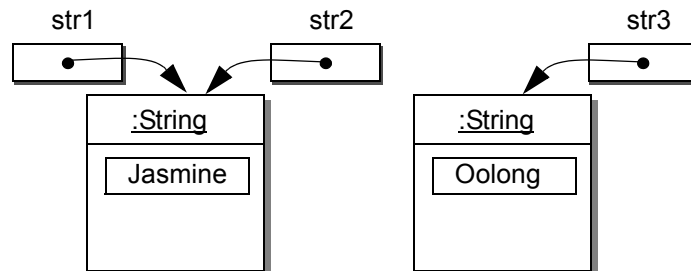
Matcher mathcer = pattern.matcher(document);

if (matcher.find()) {
    System.out.println("Found");
} else {
    System.out.println("Not found");
}
```

9.5 Comparing Strings

1. Show the state of memory after the following statements are executed:

```
String str1, str2, str3;
str1 = "Jasmine";
str2 = "Oolong";
str3 = str2;
str2 = str1;
```



9.6 StringBuffer

1. Determine the value of `str` after the following statements are executed:

a.

```
StringBuffer str
    = new StringBuffer( "Caffeine" );
str.insert(0, "Dr. ");
```

“Dr. Caffeine”

b.

```
String      str      = "Caffeine";
StringBuffer str1    =
    new StringBuffer( str.substring(1, 3) );
str1.append('e');
str = "De" + str1;
```

“Deafe”

c.

```
String      str      = "Caffeine";
StringBuffer str1    =
    new StringBuffer( str.substring(4, 8) );
str1.insert(3, 'f');
str = "De" + str1;
```

“Deeinfē”

2. Assume a `String` object `str` is assigned as

```
String str = inputBox.getString();
```

Write a code segment to replace all occurrences of lowercase vowels in a given string to the letter C by using `String` and `StringBuffer` objects.

```
StringBuffer strBuf = new StringBuffer( "" );
int max = str.length();
char letter;

for (int i = 0; i < max; i++) {
    letter = str.charAt( i );

    if (letter == 'a' || letter == 'e' ||
        letter == 'i' || letter == 'o' ||
```

```

letter == 'u' ) {
    strBuf.append('C');
}
else {
    strBuf.append( letter );
}
}
str = strBuf.toString();

```

3. Find the errors in the following code:

```

String      str      = "Caffeine";
1 —▶ StringBuffer str1 = str.substring(1, 3);
           str1.append('e');
2 —▶ System.out(str1);
           str1 = str1 + str;

```

1. *Cannot assign a String value to a StringBuffer variable.*

2. *Method print or println is missing.*

9.7 Sample Development: Building Word Concordance

No Quick Check Questions.

