

1. Suppose that s1, s2, s3, and s4 are four strings, given as follows:

String s1 = "Welcome to Java";

String s2 = s1;

String s3 = new String("Welcome to Java");

String s4 = "Welcome to Java";

What are the results of the following expressions?

a. s1 == s2

b. s1 == s3

c. s1 == s4

d. s1.equals(s3)

e. s1.equals(s4)

f. "Welcome to Java".replace("Java", "HTML")

g. s1.replace('o', 'T')

h. s1.replaceAll("o", "T")

i. s1.replaceFirst("o", "T")

j. s1.toCharArray()

a. true

b. false

c. true

d. true

e. true

f. Welcome to HTML

g. WelcTme tT Java

h. WelcTme tT Java

i. WelcTme to Java

j. toCharArray() returns an array of characters consisting of W, e, l, c, o, m, e, , t, o, , J, a, v, a

2. To create the string Welcome to Java, you may use a statement like this:

String s = "Welcome to Java";

or:

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

Which one is better? Why?

String s = "Welcome to Java";

is better, because this type of string is stored as an interned string. The interned strings of the same value share the same object.

3.What is the output of the following code?

String s1 = "Welcome to Java";

String s2 = s1.replace("o", "abc");

System.out.println(s1);

System.out.println(s2);

The output is

Welcome to Java

Welcabcmecme tabc Java

Hint: No method in the String class can change the content of the string. String is an immutable class.

4. Let s1 be " Welcome " and s2 be " welcome ". Write the code for the following statements:

- a. Replace all occurrences of the character e with E in s1 and assign the new string to s3.
- b. Split Welcome to Java and HTML into an array tokens delimited by a space and assign the first two tokens into s1 and s2.

a.

```
String s3 = s1.replace('e', 'E');
```

b.

```
String[] tokens = "Welcome to Java and HTML".split(' ');
```

```
s1 = tokens[0];
```

```
s2 = tokens[1];
```

Does any method in the String class change the contents of the string?

No

Suppose string s is created using new String(); what is s.length()?

0

How do you convert a char, an array of characters, or a number to a string?

Use the overloaded static valueOf method in the String class.

Why does the following code cause a NullPointerException?

```
1 public class Test {
2     private String text;
3
4     public Test(String s) {
5         String text = s;
6     }
7
8     public static void main(String[] args) {
9         Test test = new Test("ABC");
10        System.out.println(test.text.toLowerCase());
11    }
12 }
```

The text is declared in Line 2 as a data field, but redeclared in Line 5 as a local variable. The local variable is assigned with the string passed to the constructor, but the data field is still null. In Line 10, test.text is null, which causes NullPointerException when invoking the toLowerCase() method.

What is wrong in the following program?

```
1 public class Test {
2     String text;
3
4     public void Test(String s) {
5         text = s;
6     }
7
8     public static void main(String[] args) {
9         Test test = new Test("ABC");
10        System.out.println(test);
11    }
12 }
```

```
11 }
12 }
```

The constructor is defined incorrectly. It should not have void.

Show the output of the following code.

```
public class Test {
    public static void main(String[] args) {
        System.out.println("Hi, ABC, good".matches("ABC "));
        System.out.println("Hi, ABC, good".matches(".*ABC.*"));
        System.out.println("A,B;C".replaceAll(";", "#"));
        System.out.println("A,B;C".replaceAll("[,;]", "#"));

        String[] tokens = "A,B;C".split("[,;]");
        for (int i = 0; i < tokens.length; i++)
            System.out.print(tokens[i] + " ");
    }
}
false
true
A,B;C
A#B#C
A B C
```

Show the output of the following code.

```
public class Test {
    public static void main(String[] args) {
        String s = "Hi, Good Morning";
        System.out.println(m(s));
    }

    public static int m(String s) {
        int count = 0;
        for (int i = 0; i < s.length(); i++)
            if (Character.isUpperCase(s.charAt(i)))
                count++;

        return count;
    }
}
3
```

What is the difference between StringBuilder and StringBuffer?

The StringBuilder class, introduced in JDK 1.5, is similar to StringBuffer except that the update methods in StringBuffer are synchronized.

How do you create a string builder from a string? How do you return a string from a string builder?

Use the StringBuilder's constructor to create a string buffer for a string, and use the toString method in StringBuilder class to return a string from a StringBuilder.

Write three statements to reverse a string s using the reverse method in the StringBuilder class.

```
StringBuilder sb = new StringBuilder(s);
```

```
sb.reverse();
```

```
s = sb.toString();
```

Write three statements to delete a substring from a string s of 20 characters, starting at index 4 and ending with index 10. Use the delete method in the StringBuilder class.

```
StringBuilder sb = new StringBuilder(s);
```

```
sb.delete(4, 11);
```

```
s = sb.toString();
```

Suppose that s1 and s2 are given as follows:

```
StringBuilder s1 = new StringBuilder("Java");
```

```
StringBuilder s2 = new StringBuilder("HTML");
```

Show the value of s1 after each of the following statements. Assume that the statements are independent.

a. s1.append(" is fun");

b. s1.append(s2);

c. s1.insert(2, "is fun");

d. s1.insert(1, s2);

e. s1.charAt(2);

f. s1.length();

g. s1.deleteCharAt(3);

h. s1.delete(1, 3);

i. s1.reverse();

j. s1.replace(1, 3, "Computer");

k. s1.substring(1, 3);

l. s1.substring(2);

(a) Java is fun

(b) JavaHTML

(c) Jais funva

(d) JHTMLava

(e) v

(f) 4

(g) Jav

(h) Ja

(i) avaJ

(j) JComputera

(k) av

(l) va

Show the output of the following program:

```
public class Test {
```

```
public static void main(String[] args) {  
    String s = "Java";  
    StringBuilder builder = new StringBuilder(s);  
    change(s, builder);  
  
    System.out.println(s);  
    System.out.println(builder);  
}  
  
private static void change(String s, StringBuilder builder) {  
    s = s + " and HTML";  
    builder.append(" and HTML");  
}  
}
```

The output is

Java

Java and HTML

NOTE: 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, builder)` method. The original String object has not been changed. Therefore, the output from the original string is Java. Inside the method, the content of the StringBuilder object is changed to Java and HTML. Therefore, the output from builder is Java and HTML.