Quiz 10 Object-Oriented Thinking

1. ___________ is attached to the class of the composing class to denote the aggregation relationship with the composed object.

a. An empty diamond

b. A solid diamond

c. An empty oval

d. A solid oval

2. An aggregation relationship is usually represented as __________ in ___________.

a. a data field/the aggregating class

b. a data field/the aggregated class

c. a method/the aggregating class

d. a method/the aggregated class

3. Which of the following statements will convert a string s into i of int type?

a. i = Integer.parseInt(s);

b. i = (new Integer(s)).intValue();

c. i = Integer.valueOf(s).intValue();

d. i = Integer.valueOf(s);

e. i = (int)(Double.parseDouble(s));

4. Which of the following statements will convert a string s into a double value d?

a. d = Double.parseDouble(s);

b. d = (new Double(s)).doubleValue();

c. d = Double.valueOf(s).doubleValue();

d. All of the above.

5. Which of the following statements convert a double value d into a string s?

a. s = (new Double(d)).toString();

b. s = d;

c. s = new Double(d).stringOf();

d. s = String.stringOf(d);

e. s = d + "";

6. Which of the following statements are correct?

a. Integer.parseInt("12", 2);

b. Integer.parseInt(100);

c. Integer.parseInt("100");

d. Integer.parseInt(100, 16);

e. Integer.parseInt("345", 8);

7. What is the output of Integer.parseInt("10", 2)?

a. 1;

b. 2;

c. 10;

d. Invalid statement;

8. In JDK 1.5, you may directly assign a primitive data type value to a wrapper object. This is called ______________.

a. auto boxing

b. auto unboxing

c. auto conversion

d. auto casting

9. In JDK 1.5, analyze the following code.

Line 1: Integer[] intArray = {1, 2, 3};

Line 2: int i = intArray[0] + intArray[1];

Line 3: int j = i + intArray[2];

Line 4: double d = intArray[0];

a. It is OK to assign 1, 2, 3 to an array of Integer objects in JDK 1.5.

b. It is OK to automatically convert an Integer object to an int value in Line 2.

c. It is OK to mix an int value with an Integer object in an expression in Line 3.

d. Line 4 is OK. An int value from intArray[0] object is assigned to a double variable d.

10. To create an instance of BigInteger for 454, use

a. BigInteger(454);

b. new BigInteger(454);

c. BigInteger("454");

d. new BigInteger("454");

11. To create an instance of BigDecimal for 454.45, use

a. BigDecimal(454.45);

b. new BigDecimal(454.45);

c. BigDecimal("454.45");

d. new BigDecimal("454.45");

12. BigInteger and BigDecimal are immutable

a. true

b. false

13. To add BigInteger b1 to b2, you write _________.

a. b1.add(b2);

b. b2.add(b1);

c. b2 = b1.add(b2);

d. b2 = b2.add(b1);

e. b1 = b2.add(b1);

14. What is the output of the following code?

```
public class Test {
  public static void main(String[] args) {
    java.math.BigInteger x = new java.math.BigInteger("3");
    java.math.BigInteger y = new java.math.BigInteger("7");
    x.add(y);
    System.out.println(x);
  }
}
```

a. 3

b. 4

c. 10

d. 11

15. To divide BigDecimal b1 by b2 and assign the result to b1, you write _________.

a. b1.divide(b2);

b. b2.divide(b1);

c. b1 = b1.divide(b2);

d. b1 = b2.divide(b1);

e. b2 = b2.divide(b1);

16. Which of the following classes are immutable?

a. Integer

b. Double

c. BigInteger

d. BigDecimal

e. String

17. Which of the following statements are correct?

a. new java.math.BigInteger("343");

b. new java.math.BigDecimal("343.445");

c. new java.math.BigInteger(343);

d. new java.math.BigDecimal(343.445);

18. 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");

19. 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

20. 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

21. 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

22. 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

23. 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

24. 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

25. 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 &gt;= s2

d. int i = s1.length

e. s1.charAt(0) = '5'

26. 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

27. 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 compile 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.

28. 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)

29. 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()

30. 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")

31. 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()

32. __________ returns a string.

a. String.valueOf(123)

b. String.valueOf(12.53)

c. String.valueOf(false)

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

33. The following program displays __________.

```
public class Test {
  public static void main(String[] args) {
    String s = "Java";
    StringBuilder builder = 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

34. 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

35. 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&nbsp;&nbsp;Java

d. Welcome t&nbsp;&nbsp;Java

36. 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 false

b. true false

c. true true

d. false true

37. 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

38. 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

39. Analyze the following code.

```
class Test {
  public static void main(String[] args)  {
    StringBuilder strBuilder = new StringBuilder(4);
    strBuilder.append("ABCDE");
    System.out.println("What's strBuilder.charAt(5)? " + strBuilder.charAt(5));
  }
}
```

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

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

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

d. The program compiles and runs fine.

40. Which of the following is true?

a. You can add characters into a string builder.

b. You can delete characters from a string builder.

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

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

41. _________ returns the last character in a StringBuilder variable named strBuilder?

a. strBuilder.charAt(strBuilder.length() - 1)

b. strBuilder.charAt(strBuilder.capacity() - 1)

c. StringBuilder.charAt(strBuilder.length() - 1)

d. StringBuilder.charAt(strBuilder.capacity() - 1)

42. Assume StringBuilder strBuilder is "ABCDEFG", after invoking _________, strBuilder contains "AEFG".

a. strBuilder.delete(0, 3)

b. strBuilder.delete(1, 3)

c. strBuilder.delete(1, 4)

d. strBuilder.delete(2, 4)

43. Assume StringBuilder strBuilder is "ABCDEFG", after invoking _________, strBuilder contains "ABCRRRRDEFG".

a. strBuilder.insert(1, "RRRR")

b. strBuilder.insert(2, "RRRR")

c. strBuilder.insert(3, "RRRR")

d. strBuilder.insert(4, "RRRR")

44. Assume StringBuilder strBuilder is "ABCCEFC", after invoking _________, strBuilder contains "ABTTEFT".

a. strBuilder.replace('C', 'T')

b. strBuilder.replace("C", "T")

c. strBuilder.replace("CC", "TT")

d. strBuilder.replace('C', "TT")

e. strBuilder.replace(2, 7, "TTEFT")

45. The StringBuilder methods _____________ not only change the contents of a string builder, but also returns a reference to the string builder.

a. delete

b. append

c. insert

d. reverse

e. replace

46. The following program displays __________.

```
public class Test {
  public static void main(String[] args) {
    String s = "Java";
    StringBuilder builder = new StringBuilder(s);
    change(builder);
    System.out.println(builder);
  }

  private static void change(StringBuilder builder) {
    builder.append(" and HTML");
  }
}
```

a. Java

b. Java and HTML

c. and HTML

d. nothing is displayed