Internationalization

For more details on SUN Certifications, visit JavaScjpDumps

Q: 01 Given:
11. String test = "This is a test";
12. String[] tokens = test.split("\s");
13. System.out.println(tokens.length);
What is the result?
A. 0
B. 1
C. 4
D. Compilation fails.
E. An exception is thrown at runtime
Answer: D

Q: 02 Given:
12. System.out.format("Pi is approximately %d.", Math.PI);
What is the result?
A. Compilation fails.
B. Pi is approximately 3.
C. Pi is approximately 3.141593.
D. An exception is thrown at runtime.
Answer: D

Q: 03 Given:
33. Date d = new Date(0);
34. String ds = "December 15, 2004";
35. // insert code here
36. try {
37. d = df.parse(ds);
38. }
39. catch(ParseException e) {
40. System.out.println("Unable to parse " + ds);
41. }
42. // insert code here too
What creates the appropriate DateFormat object and adds a day to the Date object?
A. 35. DateFormat df = DateFormat.getDateFormat();
42. d.setTime((60 * 60 * 24) + d.getTime());
B. 35. DateFormat df = DateFormat.getDateInstance();
42. d.setTime((1000 * 60 * 60 * 24) + d.getTime());

C. 35. DateFormat df = DateFormat.getDateFormat();
42. d.setLocalTime((1000*60*60*24) + d.getLocalTime());
D. 35. DateFormat df = DateFormat.getDateInstance();
42. d.setLocalTime((60 * 60 * 24) + d.getLocalTime());
Answer: B

Q: 04 Given:
12. NumberFormat nf = NumberFormat.getInstance();
13. nf.setMaximumFractionDigits(4);
14. nf.setMinimumFractionDigits(2);
15. String a = nf.format(3.1415926);
16. String b = nf.format(2);
Which two statements are true about the result if the default locale is Locale.US? (Choose two.)
A. The value of b is 2.
B. The value of a is 3.14.
C. The value of b is 2.00.
D. The value of a is 3.141.
E. The value of a is 3.1415.
F. The value of a is 3.1416.
G. The value of b is 2.0000.
Answer: C, F

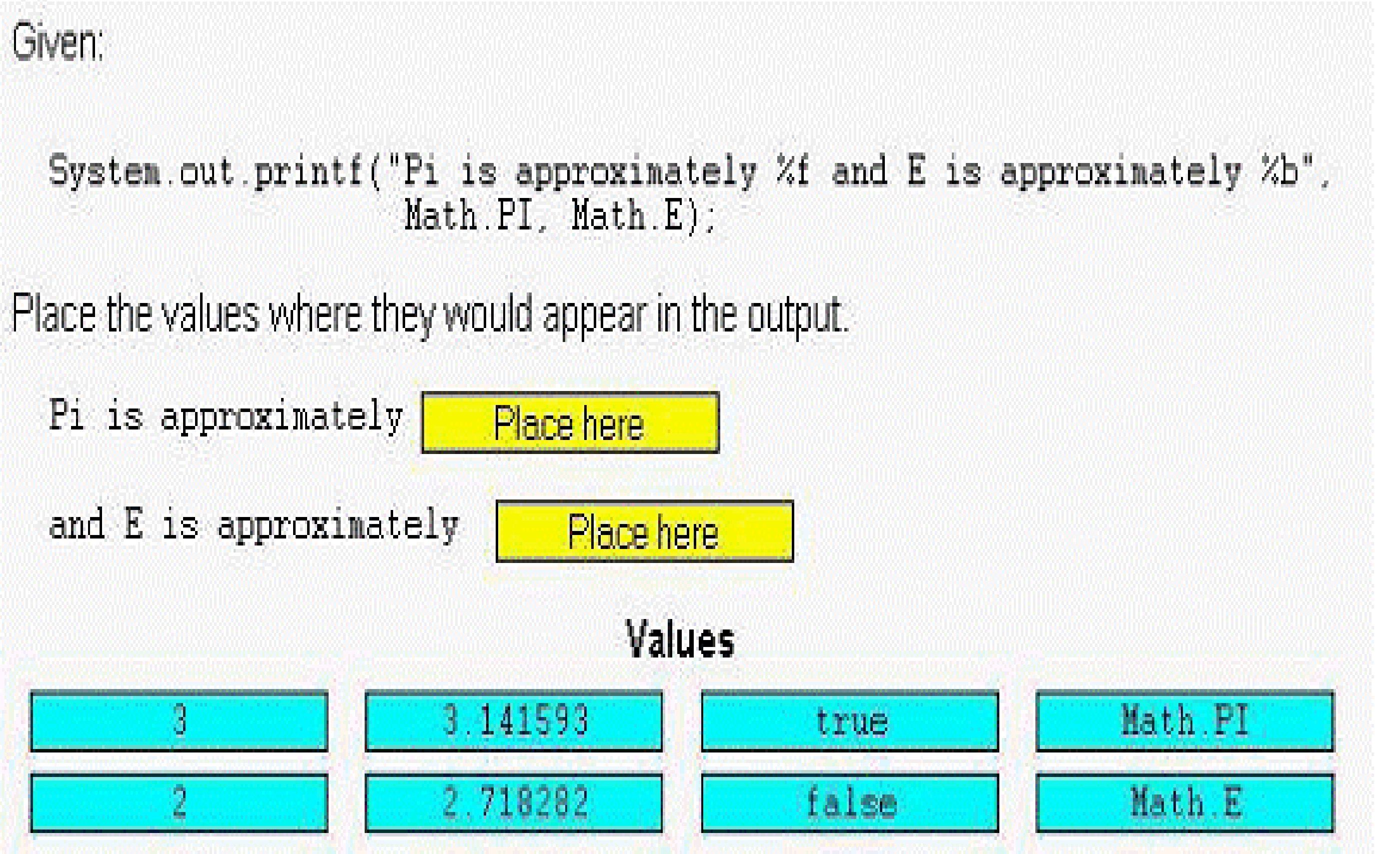
Q: 05 Given:
12. Date date = new Date();
13. df.setLocale(Locale.ITALY);
14. String s = df.format(date);
The variable df is an object of type DateFormat that has been initialized in line 11.
What is the result if this code is run on December 14, 2000?
A. The value of s is 14-dic-2004.
B. The value of s is Dec 14, 2000.
C. An exception is thrown at runtime.
D. Compilation fails because of an error in line 13.
Answer: D

Q: 06 Given:
d is a valid, non-null Date object
df is a valid, non-null DateFormat object set to the current locale
What outputs the current locale's country name and the appropriate version of d's date?
A. Locale loc = Locale.getLocale();
System.out.println(loc.getDisplayCountry()
+ " " + df.format(d));

B. Locale loc = Locale.getDefault();
System.out.println(loc.getDisplayCountry()
+ " " + df.format(d));
C. Locale loc = Locale.getLocale();
System.out.println(loc.getDisplayCountry()
+ " " + df.setDateFormat(d));
D. Locale loc = Locale.getDefault();
System.out.println(loc.getDisplayCountry()
+ " " + df.setDateFormat(d));
Answer: B

Q: 07
Given a valid DateFormat object named df, and
16. Date d = new Date(0L);
17. String ds = "December 15, 2004";
18. // insert code here
What updates d's value with the date represented by ds?
A. 18. d = df.parse(ds);
B. 18. d = df.getDate(ds);
C. 18. try {
19. d = df.parse(ds);
20. } catch(ParseException e) { };
D. 18. try {
19. d = df.getDate(ds);
20. } catch(ParseException e) { };
Answer: C

Q: 08 Click the Task button.



Given:

```
System.out.printf("Pi is approximately %f and E is approximately %b",
                  Math.PI, Math.E);
```

Place the values where they would appear in the output.

Pi is approximately

and E is approximately

Solution:
Solution:
Pi is Approximately 3.141593
and E is Approximately true

Q: 09 Given:
11. double input = 314159.26;
12. NumberFormat nf = NumberFormat.getInstance(Locale.ITALIAN);
13. String b;
14. //insert code here
Which code, inserted at line 14, sets the value of b to 314.159,26?
A. b = nf.parse(input);
B. b = nf.format(input);
C. b = nf.equals(input);
D. b = nf.parseObject(input);
Answer: B

Q: 10 Given:
12. String csv = "Sue,5,true,3";
13. Scanner scanner = new Scanner(csv);
14. scanner.useDelimiter(",");

```
15. int age = scanner.nextInt();
```

What is the result?

A. Compilation fails.

B. After line 15, the value of age is 5.

C. After line 15, the value of age is 3.

D. An exception is thrown at runtime.

Answer: D

Q: 11 Given:

```
11. String test = "a1b2c3";
12. String[] tokens = test.split("\\d");
13. for(String s: tokens) System.out.print(s + " ");
```

What is the result?

A. a b c

B. 1 2 3

C. a1b2c3

D. a1 b2 c3

E. Compilation fails.

F. The code runs with no output.

G. An exception is thrown at runtime.

Answer: A

Question: 12

Given:

```
14. DateFormat df;
15. Date date = new Date();
16. //insert code here
17. String s = df.format( date);
```

Which two, inserted independently at line 16, allow the code to compile? (Choose two.)

A. df= new DateFormat();

B. df= Date.getFormatter();

C. df= date.getFormatter();

D. df= date.getDateFormatter();

E. df= Date.getDateFormatter();

F. df= DateFormat.getInstance();

G. df = DateFormat.getDateInstance();

Answer: FG

Question: 13

Given:

```
11. String test = “Test A. Test B. Test C.”;
12. // insert code here
```

13. String[] result = test.split(regex);
Which regular expression inserted at line 12 will correctly split test into “Test A,” “Test B,” and “Test C”?

A. String regex = “”;
B. String regex = “ “;
C. String regex = “.*“.
D. String regex = “\\s”
E. String regex = “\\.\\s*”;
F. String regex = “\\w[\.] +“;

Answer: E

14. Given:

```
import java.util.regex.*;
class Regex2 {
public static void main(String[] args) {
Pattern p = Pattern.compile(args[0]);
Matcher m = p.matcher(args[1]);
boolean b = false;
while(b = m.find()) {
System.out.print(m.start() + m.group());
}
}
}
```

And the command line:

```
java Regex2 "\d*" ab34ef
```

What is the result?

A. 234 B. 334
C. 2334 D. 0123456
E. 01234456 F. 12334567
G. Compilation fails.

Answer:

-> **E** is correct. The \d is looking for digits. The * is a quantifier that looks for 0 to many occurrences of the pattern that precedes it. Because we specified *, the group() method returns empty Strings until consecutive digits are found, so the only time group() returns a value is when it returns 34 when the matcher finds digits starting in position 2. The start() method returns the starting position of the previous match because, again, we said find 0 to many occurrences.

->**A, B, C, D, E, F,** and **G** are incorrect based on the above.

15. Given:

```
1. import java.text.*;
2. class DateOne {
3. public static void main(String[] args) {
4. Date d = new Date(1123631685981L);
DateFormat df = new DateFormat();
System.out.println(df.format(d));
}
}
```

And given that 1123631685981L is the number of milliseconds between Jan. 1, 1970, and sometime on Aug. 9, 2005, what is the result? (Note: the time of day in option A may vary.)

A. 8/9/05 5:54 PM
B. 1123631685981L
C. An exception is thrown at runtime.
D. Compilation fails due to a single error in the code.
E. Compilation fails due to multiple errors in the code.

Answer:

-> **E** is correct. The Date class is located in the java.util package so it needs an import, and DateFormat objects must be created using a static method such as DateFormat.getInstance() or DateFormat.getDateInstance().
->**A, B, C,** and **D** are incorrect based on the above.

16. Which are true? (Choose all that apply.)

A. The DateFormat.getDate() is used to convert a String to a Date instance.
B. Both DateFormat and NumberFormat objects can be constructed to be Locale specific.
C. Both Currency and NumberFormat objects must be constructed using static methods.
D. If a NumberFormat instance's Locale is to be different than the current Locale, it must be specified at creation time.
E. A single instance of NumberFormat can be used to create Number objects from Strings and to create formatted numbers from numbers.

Answer:

-> **B , C, D,** and **E** are correct.
->**A** is incorrect, DateFormat.parse() is used to convert a String to a Date.

17. Which will compile and run without exception? (Choose all that apply.)

A. System.out.format("%b", 123);
B. System.out.format("%c", "x");
C. System.out.printf("%d", 123);
D. System.out.printf("%f", 123);
E. System.out.printf("%d", 123.45);
F. System.out.printf("%f", 123.45);
G. System.out.format("%s", new Long("123"));

Answer:

->**A , C, F,** and **G** are correct. The %b (boolean) conversion character returns true for any non-null or non-boolean argument.
->**B** is incorrect, the %c (character) conversion character expects a character, not a String. **D** is incorrect, the %f (floating-point) conversion character won't automatically promote an integer type. **E** is incorrect, the %d (integral) conversion character won't take a floatingpoint number.

(Note: The format() and printf() methods behave identically.)

18. Given:

```
1. import java.util.*;
2. class Brain {
3. public static void main(String[] args) {
4. // insert code block here
5. }
6. }
```

Which, inserted independently at line 4, compile and produce the output "123 82"? (Choose all that apply.)

```
A. Scanner sc = new Scanner("123 A 3b c,45, x5x,76 82 L");
while(sc.hasNextInt()) System.out.print(sc.nextInt() + " ");
B. Scanner sc = new Scanner("123 A 3b c,45, x5x,76 82 L").
useDelimiter(" ");
while(sc.hasNextInt()) System.out.print(sc.nextInt() + " ");
C. Scanner sc = new Scanner("123 A 3b c,45, x5x,76 82 L");
while(sc.hasNext()) {
if(sc.hasNextInt()) System.out.print(sc.nextInt() + " ");
else sc.next(); }
D. Scanner sc = new Scanner("123 A 3b c,45, x5x,76 82 L").
useDelimiter(" ");
while(sc.hasNext()) {
if(sc.hasNextInt()) System.out.print(sc.nextInt() + " ");
else sc.next(); }
E. Scanner sc = new Scanner("123 A 3b c,45, x5x,76 82 L");
do {
if(sc.hasNextInt()) System.out.print(sc.nextInt() + " ");
} while ( sc.hasNext() );
F. Scanner sc = new Scanner("123 A 3b c,45, x5x,76 82 L").
useDelimiter(" ");
do {
if(sc.hasNextInt()) System.out.print(sc.nextInt() + " ");
} while ( sc.hasNext() );
```

Answer:

->**C** and **D** are correct. Whitespace is the default delimiter, and the while loop advances through the String using nextInt() or next().

->**A** and **B** are incorrect because the while loop won't progress past the first non-int.

E and **F** are incorrect. The do loop will loop endlessly once the first non-int is found because hasNext() does not advance through data.