



CS101 Part 2: Practice Questions

Algorithms on Arrays, Classes and Objects, String Class, Stack Class

1. Write a method that, given two sorted arrays of integers, merges the two arrays into a single sorted array that is returned.
2. Write a method that, given an array of integers, computes the sum of every other number (starting at the zero'th index) and returns true if the sum is divisible by 10 and false otherwise.
3. What is the output of the following program lines when they are embedded in a correct Java program.

```
1 public class Test {
2     public static void main(String[] args) {
3         int number = 0;
4         int[] numbers = new int[1];
5         numbers[0] = 0;
6         m(number, numbers);
7         System.out.println("number is " + number
8             + " and numbers[0] is " + numbers[0]);
9     }
10
11    public static void m(int x, int[] y) {
12        x = 3;
13        y[0] = 3;
14    }
15 }
16
```

4. Write the following method that returns true if the list is already sorted in increasing order and false if it is not sorted.
`public static boolean isSorted(int[][] list)`
5. Write a method that given a list of floating point numbers determines if a particular value is on the list. Your method should return the location of the item if it is found, or -1 if it is not found. Use the following method header:
`public static int find(double [][] list, double key)`
Use the searching algorithm of your choice.
6. What is wrong with the following program or code fragment? How would you correct the errors?

- (a) assume that the `Circle` class has a default constructor and `getRadius` method

```
public class ShowErrors {
    public void method1() {
        Circle c;
        System.out.println("What is radius "
            + c.getRadius());
        c = new Circle();
    }
}
```

- (b)
- ```
public class ShowErrors {
 public static void main(String[] args) {
 C c = new C(5.0);
 System.out.println(c.value);
 }
}
class C {
 int value = 2;
}
```



```
(c) public class Test {
 public static void main(String[] args) {
 A a = new A();
 a.print();
 }
}

class A {
 private String s;
 A(String s) {
 this.s = s;
 }
 public void print() {
 System.out.print(s);
 }
}
```

7. If all objects of a class need to share the same variable, how must you declare that variable? Show an example of such definition.
8. To make a data field or a method accessible by only other members of its class, what access modifier must be used?
9. Can you invoke an instance method or reference an instance variable from a static method? Can you invoke a static method or reference a static variable from an instance method?
10. What is the difference between a class and an object?
11. What does each object have its own copy of?
12. What name does a constructor have?
13. What is wrong with the following program or code fragment? How would you correct the errors?

(a) assume that the **Circle** class has a default constructor

```
public class Foo {
 public static void method0() {
 method1();
 }
 public void method1() {
 method2();
 }
 public static void method2() {
 System.out.println("What is radius " + c.getRadius());
 }
 Circle c = new Circle();
}
```

(b) `public class C {`

```
 private int p;

 public C() {
 C(0);
 }
 public C(int p) {
 p = p;
 }
 public void setP(int p) {
 p = p;
 }
}
```



14. What does the **new** operator do?
15. What is the **this** reference used for? Describe all of its uses.
16. Can a constructor have one or more parameters? Can a class have more than one constructor?
17. Describe the difference between passing a parameter of a primitive type and passing a parameter of a reference type. Give examples of both the primitive types and the reference types.
18. Write a method that, given an array of **Circle** objects, sorts the circles from smallest to largest. Assume that the **Circle** class is defined as follows:

```
public class Circle {
 private float radius;
 public Circle (float r) {
 if (r >=0) radius = r;
 else radius = 1;
 }
 public float getRadius() {
 return radius;
 }
 public int compareTo(Circle c) {
 if (radius == c.radius) return 0;
 else if (radius < c.radius) return -1;
 else return 1;
 }
}
```

You should use the following signature for the method:

```
void sortCircles (Circle [] circles)
```

You can use any sorting algorithm we discussed.

19. Suppose that **s1** and **s2** are two instances of the Java **String** class. Which of the following statements or expressions are incorrect?

- (a) `String s = new String("new string");`
- (b) `String s3 = s1 + s2;`
- (c) `String s3 = s1 - s2;`
- (d) `s1 == s2;`
- (e) `s1 >= s2;`
- (f) `s1.compareTo(s2);`
- (g) `int i = s1.length();`
- (h) `char c = s1(0);`
- (i) `char c = s1.charAt(s1.length());`

20. Suppose that the class **Foo** is defined as shown. Let **f** be an instance of **Foo**. Which of the statements below are invalid?

```
public class Foo {
 int i;
 static String s;
 void imethod() { ... }
 static void smethod() { ... }
}
```



- (a) `System.out.println(f.i);`
- (b) `System.out.println(f.s);`
- (c) `f.imethod();`
- (d) `f.smethod();`
- (e) `System.out.println(Foo.i);`
- (f) `System.out.println(Foo.s);`
- (g) `Foo.imethod();`
- (h) `Foo.smethod();`

21. Write a method that given a string object that consists of digits, computes and returns the sum of every other digit (starting at the very first one). The string of digits can be arbitrary length, but you may assume that the sum of the digits fits in a variable of type `int`.
22. Write the Java function `lengthOfSortedSequence` which returns the length of the longest sorted sequence in an array. For this problem a sorted sequence is a sequence of non-decreasing values. Here are some examples:

| array                                        | return value from <code>lengthOfSortedSequence(array)</code> |
|----------------------------------------------|--------------------------------------------------------------|
| <code>[-7 3 99 -10 0 0 43 10 20 30 5]</code> | 4                                                            |
| <code>[-1, 2, 3, 3]</code>                   | 4                                                            |
| <code>[5, 2, 1]</code>                       | 1                                                            |
| <code>[1]</code>                             | 1                                                            |
| <code>[]</code>                              | 0                                                            |

23. In the following class, the variable `n` occurs in multiple scopes. Which declarations of `n` are legal and which are illegal?

```

1 public class X
2 {
3 private int n;
4
5 public int f()
6 {
7 int n = 1;
8 return n;
9 }
10
11 public int g(int k)
12 {
13 int a;
14 for (int n = 1; n <= k; n++)
15 a = a + n;
16 return a;
17 }
18 public int h(int n)
19 {
20 int b;
21 for (int n = 1; n <= 10; n++)
22 b = b + n;
23 return b + n;
24 }
25
26 public int k(int n)
27 {
28 if (n < 0)
29 {
30 int k = -n;
31 int n = (int) (Math.sqrt(k));
32 return n;

```



```

33 }
34 else return n;
35 }
36
37 public int m(int k)
38 {
39 int a;
40 for (int n = 1; n <= k; n++)
41 a = a + n;
42 for (int n = k; n >= 1; n++)
43 a = a + n;
44 return a;
45 }
46 }

```

24. Show the output of the following programs:

(a)

```

1 public class Test {
2 public static void main(String[] args) {
3 Count myCount = new Count();
4 int times = 0;
5 for (int i = 0; i < 100; i++)
6 increment(myCount, times);
7 System.out.println("count is " + myCount.count);
8 System.out.println("times is " + times);
9 }
10
11 public static void increment(Count c, int times) {
12 c.count++;
13 times++;
14 }
15 }

```

```

1 public class Count {
2 public int count;
3 public Count(int c) {
4 count = c;
5 }
6 public Count() {
7 count = 1;
8 }
9 }
10
11
12
13
14
15

```

(b)

```

1 public class Test {
2 public static void main(String[] args) {
3 T t1 = new T();
4 T t2 = new T();
5 System.out.println("t1's i = " +
6 t1.i + " and j = " + t1.j);
7 System.out.println("t2's i = " +
8 t2.i + " and j = " + t2.j);
9 }
10 }

```

```

1 class T {
2 static int i = 0;
3 int j = 0;
4 T() {
5 i++;
6 j = 1;
7 }
8 }
9
10

```

(c)

```

1 public class Test {
2 public static void main(String[] args) {
3 int[] a = {1, 2};
4 swap(a[0], a[1]);
5 System.out.println("a[0] = " + a[0]
6 + " a[1] = " + a[1]);
7 }
8 public static void swap(int n1, int n2) {
9 int temp = n1;
10 n1 = n2;
11 n2 = temp;
12 }
13 }

```



(d)

```
1 public class Test {
2 public static void main(String[] args) {
3 int[] a = {1, 2};
4 swap(a);
5 System.out.println("a[0] = " + a[0]
6 + " a[1] = " + a[1]);
7 }
8 public static void swap(int[] a) {
9 int temp = a[0];
10 a[0] = a[1];
11 a[1] = temp;
12 }
13 }
```

25. Consider the following code.

```
1 public class Ball
2 {
3 private double rx, ry;
4 public Ball()
5 {
6 rx = 0.5;
7 ry = 0.5;
8 }
9 public void move()
10 {
11 rx = rx + .001;
12 ry = ry + .002;
13 }
14 public void move(double vx, double vy)
15 {
16 rx = rx + vx;
17 ry = ry + vy;
18 }
19 ...
20 public static void main(String[] args)
21 {
22 private double vx, vy;
23 Ball b1 = new Ball();
24 Ball b2 = new Ball();
25 b1.move();
26 ...
27 b2.move(vx, vy);
28 ...
29 }
30 }
```

In the blanks at left, give all line numbers that contain each of the entities described at right.

- (a) .....instance variable declaration
- (b) .....constructor signature
- (c) .....object creation
- (d) .....overloaded method signature
- (e) .....method invocation or call
- (f) .....primitive type variable declaration
- (g) .....reference type variable declaration

26. What is printed?

- (a) Show the output of the following program:



```
1 public class Test {
2 public static void main(String[] args) {
3 Counter myCounter = new Counter();
4 int times = 0;
5
6 for (int i = 0; i < 100; i++)
7 increment(myCounter, times);
8
9 System.out.println("counter is " + myCounter.count);
10 System.out.println("times is " + times);
11 }
12
13 public static void increment(Counter c, int times) {
14 c.count++;
15 times++;
16 }
17 }
18
19 public class Counter {
20 public int count;
21
22 Counter() {
23 count = 0;
24 }
25 }
```

(b) Show the output of the following program:

```
1 public class Test {
2 public static void main(String[] args) {
3 int[][] array = {{1, 2, 3 }, {4, 5, 6},
4 { 7, 8, 9}, {10, 11, 12} };
5 System.out.println(array.length);
6 System.out.println(array[0].length);
7 }
8 }
```

27. Write a method that given a two dimensional array of real numbers creates and returns a one dimensional that contains sums of all the rows.
28. Write a method that given a two dimensional array of real numbers creates and returns a one dimensional that contains sums of all the columns.
29. Write a method that given a square two dimensional array (number of columns is equal to the number of rows) prints to the screen values that appear on the diagonal of the array.
30. Write a method that given a two dimensional array returns either true, if the array has the same number of columns in each row, or false, if the array is ragged (not all rows have the same number of columns).
31. The class contract for StackOfCharacters class that we discussed is provided below. Remember that the capacity of the stack is doubled every time the size of the stack equals the capacity of the stack.



**StackOfCharacters:**

|                                         |                                                                                                                                                            |
|-----------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>StackOfCharacters()</b>              | Constructs an empty stack of characters with a default capacity of 16.                                                                                     |
| <b>StackOfCharacters(capacity: int)</b> | Constructs an empty stack of characters with a specified capacity (if provided capacity is less than or equal to zero, the capacity is set to default 16). |
| <b>empty(): boolean</b>                 | Returns true if the stack is empty, otherwise it returns false.                                                                                            |
| <b>getSize(): int</b>                   | Returns the number of elements in the stack.                                                                                                               |
| <b>getCapacity(): int</b>               | Returns the capacity of the stack.                                                                                                                         |
| <b>push(value: Character): void</b>     | Puts the character value at the top of the stack.                                                                                                          |
| <b>pop(): Character</b>                 | Removes the character from the top of the stack and returns it. If stack is empty, returns null.                                                           |
| <b>peek(): Character</b>                | Returns the character from the top of the stack (without removing it). If stack is empty, returns null.                                                    |

What is the output of the following program that uses StackOfCharacters class?

```
1 public class TestingStack {
2 public static void main(String [] args) {
3 StackOfCharacters s1 = new StackOfCharacters(2);
4 StackOfCharacters s2 = new StackOfCharacters();
5
6 System.out.println(s1.getSize());
7 System.out.println(s2.getSize());
8
9 s1.push('a');
10 s1.push('b');
11 s1.push('c');
12 s2.push('w');
13 s2.push('x');
14 s2.push('y');
15
16 System.out.println(s1.getSize());
17 System.out.println(s2.getSize());
18
19 System.out.println(s2.pop());
20 System.out.println(s2.pop());
21 System.out.println(s2.peek());
22 s2.pop();
23
24 if (! s2.empty())
25 System.out.println("not empty");
26
27 }
```