

APCSA Practice Questions

Directions: Circle the correct response.

1. Evaluate the following expression: $4 + 6 \% 12 / 4$

- (A) 1
- (B) 2
- (C) 4
- (D) 4.5
- (E) 5

2. Determine the output of the following code:

```
String animal1 = "elephant";
String animal2 = "lion";
swap(animal1, animal2);
animal1.toUpperCase();
animal2.toLowerCase();

System.out.println(animal1 + " " + animal2);

public static void swap(String a1, String a2) {
    String hold = a1;
    a1 = a2;
    a2 = hold;
}
```

- (A) elephant lion
- (B) ELEPHANT lion
- (C) lion elephant
- (D) LION elephant
- (E) LION ELEPHANT

3. A programmer intended to write code to print three words in ascending lexicographical order. Follow the code and determine the printed output.

```
1  String word1 = "frog";
2  String word2 = "dog";
3  String word3 = "cat";
4
5  if (word1.compareTo(word2) < 0)
6      if (word2.compareTo(word3) < 0)
7          System.out.println(word1 + " " + word2 + " " + word3);
8      else
9          System.out.println(word1 + " " + word3 + " " + word2);
10 else
11     if (word1.compareTo(word2) > 0)
12         if (word2.compareTo(word3) < 0)
13             System.out.println(word1 + " " + word2 + " " + word3);
14         else
15             System.out.println(word1 + " " + word3 + " " + word2);
16     else
17         if (word2.equals(word3))
18             System.out.println("all the words are the same");
19         else
20             System.out.println("word1 and word2 are duplicates");
```

- (A) frog cat dog
- (B) cat dog frog
- (C) dog frog cat
- (D) frog dog cat
- (E) dog cat frog

4. Choose the correct option to complete lines 3 and 4 such that str2 will contain the letters of str1 in reverse order.

```
1  String str1 = "banana";
2  String str2 = "";
3  // missing code
4  // missing code
5  {
6      str2 += str1.substring(i, i + 1);
7      i--;
8  }
```

- (A) int i = 0;
 while (i < str1.length)
- (B) int i = str1.length();
 while (i >= 0)
- (C) int i = str1.length() - 1;
 while (i >= 0)
- (D) int i = str1.length();
 while (i > 0)
- (E) int i = str1.length() - 1;
 while (i > 0)

5. Refer to the following code for the Tile class below to respond to questions 6 and 7.

```
1  public class Tile
2  {
3      private int styleNumber;
4      private String color;
5      private double width;
6      private double height;
7      private String material;
8      private double price;

9      Tile(int style, String col)
10     {
11         styleNumber = style;
12         color = col;
13     }
14     Tile(int style, String col, double w, double h, String mat, double price)
15     {
16         styleNumber = style;
17         color = col;
18         width = w;
19         height = h;
20         material = mat;
21         price = price;
22     }
23     Tile(int style, String col, String mat, double price)
24     {
25         styleNumber = style;
26         color = col;
27         material = mat;
28         price = price;
29     }
30     public void chgMaterial(String mat)
31     {
32         String material = mat;
33     }
34     public String toString()
35     {
36         return (styleNumber + " " + color + " " + width + " " + height + " " +
37             material + " " + price);
38     }
39 }
```

What is the output after the following code is executed?

```
Tile t1 = new Tile(785, "grey", "ceramic", 6.95);t1.chgMaterial("marble");
System.out.print(t1.toString());
```

- (A) Tile@5ccd43c2
- (B) 785 grey 0.0 0.0 marble 0.0
- (C) 785 grey 0.0 0.0 ceramic 0.0
- (D) 785 grey 0.0 0.0 ceramic 6.95
- (E) 785 grey 0.0 0.0 marble 6.95

6. The Tile class is going to be used for an application built for a small independent tile store. The owner wants the programmer to add a field for the number of unopened boxes of tile he has for each style of tile he has in stock and a method to change the value. What would be the proper declaration for this field?

(A) public static int inventory;
(B) private static double inventory;
(C) final int inventory;
(D) private int inventory;
(E) private int [] inventory;

7. Given nums—a rectangular, two-dimensional array of integers—choose the code to print the entire array.

I.	<pre>for (int r = 0; r < nums.length; r++) { for (int c = 0; c < nums[0].length; c++) { System.out.print(nums[r][c]); } System.out.print("\n"); }</pre>
II.	<pre>for (int [] row : nums) { for (int col : row) { System.out.print(col + " "); } System.out.println(""); }</pre>
III.	<pre>for (int r = 0; r < nums[0].length; r++) { for (int c = 0; c < nums.length; c++) { System.out.print(nums[r][c] + " "); } System.out.print("\n"); }</pre>

(A) I only
(B) I and II only
(C) I and III only
(D) II and III only
(E) I, II, and III

Refer to the Percussion class and Xylophone class below to answer questions 8 and 9.

```
public class Percussion {
    private String name;
    private double weight;
    Percussion() {
    }
    Percussion(String n, double w)
    {
        name = n;
        weight = w;
    }
    public String getName()
    {
        return name;
    }
    public double getWeight()
    {
        return weight;
    }
}
public class Drums extends Percussion
{
}
public class Xylophone extends Percussion {
    private int numberOfKeys;

    Xylophone(String name, double weight, int numberOfKeys){

        <missing code>

    }
    public int getNumKeys()
    {
        return numberOfKeys;
    }
}
```

8. Which is the most appropriate replacement for <missing code> in the Xylophone constructor?

(A)	<pre>this.numberOfKeys = numberOfKeys; super(name, weight);</pre>
(B)	<pre>super(name, weight); this.numberOfKeys = numberOfKeys;</pre>
(C)	<pre>super(name, weight); numberOfKeys = this.numberOfKeys;</pre>
(D)	<pre>this.numberOfKeys = numberOfKeys;</pre>
(E)	<pre>numberOfKeys = this.numberOfKeys;</pre>

9. A client program wishes to compare the two xylophone objects as follows:
Xylophone x2 = new Xylophone ("xylophone", 80, 32);
Xylophone x3 = new Xylophone ("xylophone", 65, 32);

The two objects should be considered “equally heavy” if and only if they have the same weight. Which of the following code excerpts accomplishes that task?

(A)	<pre>if (x2.weight == x3.weight) System.out.println("equally heavy"); else System.out.println("not equally heavy");</pre>
(B)	<pre>if (x2.weight() == x3.weight()) System.out.println("equally heavy"); else System.out.println("not equally heavy");</pre>
(C)	<pre>if (x2.getWeight() == x3.getWeight()) System.out.println("equally heavy"); else System.out.println("not equally heavy");</pre>
(D)	<pre>if (x2.weight.equals(x3.weight)) System.out.println("equally heavy"); else System.out.println("not equally heavy");</pre>
(E)	The weights of the objects cannot be compared.

10. Choose the best solution to complete the missing code such that the code will implement a binary search to find the variable number in arr.

```
int number = <some number in arr>;
System.out.println(search(arr, 0, arr.length - 1, number));

public int search(int[] a, int first, int last, int sought) {
    int mid = (first + last) / 2;

    if (<missing code>) {
        last = mid - 1;
        return search(a, first, last, sought);
    }
    else if (<missing code>)) {
        first = mid + 1;
        return search(a, first, last, sought);
    }

    return mid;
}
```

- | | |
|---------------------------|-----------------------|
| (A) $a[mid] > sought$ | $a[mid] < sought$ |
| (B) $a[mid] + 1 > sought$ | $a[mid] < sought$ |
| (C) $a[mid] > sought$ | $a[mid] - 1 < sought$ |
| (D) $a[mid] + 1 > sought$ | $a[mid] - 1 < sought$ |
| (E) $a[mid] = sought$ | $a[mid] = sought$ |

11. A mountain climbing club maintains a record of the climbs that its members have made. Information about a climb includes the name of the mountain peak and the amount of time it took to reach the top. The information is contained in the `ClimbInfo` class as declared below.

```
public class ClimbInfo
{
    /** Creates a ClimbInfo object with name peakName and time climbTime.
     * @param peakName the name of the mountain peak
     * @param climbTime the number of minutes taken to complete the climb
     */
    public ClimbInfo(String peakName, int climbTime)
    { /* implementation not shown */ }

    /** @return the name of the mountain peak
     */
    public String getName()
    { /* implementation not shown */ }

    /** @return the number of minutes taken to complete the climb
     */
    public int getTime()
    { /* implementation not shown */ }

    // There may be instance variables, constructors, and methods that are not shown.
}
```

The `ClimbingClub` class maintains a list of the climbs made by members of the club. The declaration of the `ClimbingClub` class is shown below. You will write two different implementations of the `addClimb` method. You will also answer two questions about an implementation of the `distinctPeakNames` method.


```

public class ClimbingClub
{
    /** The list of climbs completed by members of the club.
     *  Guaranteed not to be null. Contains only non-null references.
     */
    private List<ClimbInfo> climbList;

    /** Creates a new ClimbingClub object. */
    public ClimbingClub()
    { climbList = new ArrayList<ClimbInfo>(); }

    /** Adds a new climb with name peakName and time climbTime to the list of climbs.
     *  @param peakName the name of the mountain peak climbed
     *  @param climbTime the number of minutes taken to complete the climb
     */
    public void addClimb(String peakName, int climbTime)
    {
        /* to be implemented in part (a) with ClimbInfo objects in the order they were added */
        /* to be implemented in part (b) with ClimbInfo objects in alphabetical order by name */
    }

    /** @return the number of distinct names in the list of climbs */
    public int distinctPeakNames()
    {
        /* implementation shown in part (c) */
    }

    // There may be instance variables, constructors, and methods that are not shown.
}

```

(a) Write an implementation of the `ClimbingClub` method `addClimb` that stores the `ClimbInfo` objects in the order they were added. This implementation of `addClimb` should create a new `ClimbInfo` object with the given name and time. It appends a reference to that object to the end of `climbList`. For example, consider the following code segment.

```

ClimbingClub hikerClub = new ClimbingClub();
hikerClub.addClimb("Monadnock", 274);
hikerClub.addClimb("Whiteface", 301);
hikerClub.addClimb("Algonquin", 225);
hikerClub.addClimb("Monadnock", 344);

```

When the code segment has completed executing, the instance variable `climbList` would contain the following entries.

Peak Name	"Monadnock"	"Whiteface"	"Algonquin"	"Monadnock"
Climb Time	274	301	225	344

Information repeated from the beginning of the question

```

public class ClimbInfo

public ClimbInfo(String peakName, int climbTime)
public String getName()
public int getTime()

public class ClimbingClub

private List<ClimbInfo> climbList
public void addClimb(String peakName, int climbTime)
public int distinctPeakNames()

```

Complete method `addClimb` below.

```

/** Adds a new climb with name peakName and time climbTime to the list of climbs.
 * @param peakName the name of the mountain peak climbed
 * @param climbTime the number of minutes taken to complete the climb
 * Postcondition: The new entry is at the end of climbList;
 * The order of the remaining entries is unchanged.
 */
public void addClimb(String peakName, int climbTime)

```

(b) Write an implementation of the `ClimbingClub` method `addClimb` that stores the elements of `climbList` in alphabetical order by name (as determined by the `compareTo` method of the `String` class). This implementation of `addClimb` should create a new `ClimbInfo` object with the given name and time and then insert the object into the appropriate position in `climbList`. Entries that have the same name will be grouped together and can appear in any order within the group. For example, consider the following code segment.

```
ClimbingClub hikerClub = new ClimbingClub();  
hikerClub.addClimb("Monadnock", 274);  
hikerClub.addClimb("Whiteface", 301);  
hikerClub.addClimb("Algonquin", 225);  
hikerClub.addClimb("Monadnock", 344);
```

When the code segment has completed execution, the instance variable `climbList` would contain the following entries in either of the orders shown below.

Peak Name	"Algonquin"	"Monadnock"	"Monadnock"	"Whiteface"
Climb Time	225	344	274	301

OR

Peak Name	"Algonquin"	"Monadnock"	"Monadnock"	"Whiteface"
Climb Time	225	274	344	301

You may assume that `climbList` is in alphabetical order by name when the method is called. When the method has completed execution, `climbList` should still be in alphabetical order by name.

Information repeated from the beginning of the question

```
public class ClimbInfo
```

```
public ClimbInfo(String peakName, int climbTime)
public String getName()
public int getTime()
```

```
public class ClimbingClub
```

```
private List<ClimbInfo> climbList
public void addClimb(String peakName, int climbTime)
public int distinctPeakNames()
```

Complete method addClimb below.

```
/** Adds a new climb with name peakName and time climbTime to the list of climbs.
 * Alphabetical order is determined by the compareTo method of the String class.
 * @param peakName the name of the mountain peak climbed
 * @param climbTime the number of minutes taken to complete the climb
 * Precondition: entries in climbList are in alphabetical order by name.
 * Postcondition: entries in climbList are in alphabetical order by name.
 */
public void addClimb(String peakName, int climbTime)
```