

MORE MULTIPLE CHOICE PRACTICE - Inheritance / Arrays

#1 Consider the following code segment.

```
int[] nums = new int[51];

for (int k = 0; k < nums.length; k++)
    nums[k] = 1;

for (int k = 3; k <= 50; k += 3)
    nums[k] = 0;

for (int k = 5; k <= 50; k += 5)
    nums[k] = 0;
```

- 14 A
- 15 D
- 31 E
- 32 C
- 35 B
- 36 B
- 13 A

How many elements in the array nums have the value 0 after this code has been executed?

- (A) 23
- (B) 25
- (C) 26
- (D) 27
- (E) 28

Handwritten analysis for Question #1:

0 1 2 3 4 50

{ 1, 1, 1, 1, 1, 1, 1, ... } length = 51 - 3

{ 1, 1, 1, 0, 1, 1, 0, 1, 1, 0, 1, 1, ... } $48 \div 3 = 16$

5 → 50 every 5th so 10

{ 1, 1, 1, 0, 1 } 0, 0, 1, 1, 0, 0, 1, 1, ...

15th, 30th, 45th counted twice so $26 - 3 = 23$

#2 Consider the following method.

```
public int countSomething(int[] p)
{
    int count = 0;
    for (int i = 0; i < p.length; i++)
    {
        count++;
        int j = p[i];
        while (j != i)
        {
            j = p[j];
            count++;
        }
    }
    return count;
}
```

Given

```
int[] arr = {0, 2, 3, 1};
```

what will countSomething(arr) return?

- (A) 2
- (B) 3
- (C) 5
- (D) 10
- (E) 13

count	j	i
1	0	0
2	2	1
3	3	2
4	1	3
5	3	
6	1	
7	2	
8	1	
9	2	
10	3	

#3

Consider the following method:

```
public int[][] makeCounts(int n)
{
    int[][] counts = new int[3][n];
    counts[0][0] = 0;
    counts[1][0] = 0;
    counts[2][0] = 1;
    for (int k = 1; k < n; k++)
    {
        counts[0][k] = counts[0][k-1] + counts[1][k-1];
        counts[1][k] = counts[1][k-1] + counts[0][k-1] +
            counts[2][k-1];
        counts[2][k] = counts[2][k-1] + counts[1][k-1];
    }
    return counts;
}
```

What values are in the array returned by makeCounts (5)?

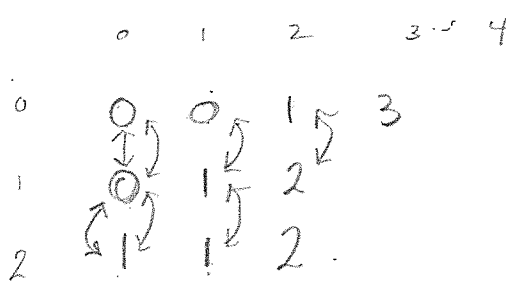
- (A)
0 0 1 1 1
0 1 1 1 1
1 1 1 1 1

- (B)
0 0 1 2 3
0 1 2 3 4
1 2 3 4 5

- (C)
0 0 1 3 6
0 1 2 5 8
1 1 4 7 9

- (D)
0 0 1 3 8
0 1 2 5 12
1 1 2 4 9

- (E)
0 1 3 9 27
0 1 3 9 27
1 1 3 9 27



#4 Consider the following method.

```
public int countSomething(int[] arr)
{
    int m = arr[0];
    int count = 1;
    for (int k = 1; k < arr.length; k++)
    {
        int a = arr[k];
        if (a > m)
        {
            m = a;
            count = 1;
        }
        else if (m == a)
            count++;
    }
    return count;
}
```

returns # of times
a max value occurs
If new max,
count = 1

For which of the following arrays countSomething will return 3?

- (A) int[] arr = {0, 1, 1, 1, 1};
- (B) int[] arr = {1, 6, 5, 4, 0};
- (C) int[] arr = {1, 0, 5, 6, 1};
- (D) int[] arr = {3, 2, 1, 0, 5};
- (E) None of the above

Animal
↓
Mammal

#13 The statement

```
Animal a = new Mammal("Elephant");
```

compiles with no errors. Which of the following situations will permit that?

- (A) Mammal is a class with a constructor that takes one parameter of the String type, and Animal is its subclass.
- (B) Animal is a class with a constructor that takes one parameter of the String type, Mammal is its subclass that has no constructors defined.
- (C) Mammal is a class with a constructor that takes one parameter of the String type, Animal is an interface, and Mammal implements Animal.
- (D) Animal has a public static data field String Mammal.
- (E) None of the above

Animal
↓
Mammal

Questions 5-6 refer to the following class `Game` and the incomplete class `ChessGame`.

```
public class Game
{
    private String gameName;
    private List<String> players;

    public Game(String name)
    {
        gameName = name;
        players = new ArrayList<String>();
    }

    public Game(String name, String[] people)
    {
        gameName = name;
        players = new ArrayList<String>();
        for (String nm : people)
            players.add(nm);
    }

    public void addPlayer(String name) { players.add(name); }

    public String getPlayer(int k)
    {
        return players.get(k - 1);
    }

    public String toString()
    {
        return gameName + " game " + players.toString();
    }
}

public class ChessGame extends Game
{
    public ChessGame(String white, String black)
    {
        < missing code >
    }
}
```

#5 Consider the following code segment in a Game's client class.

```
String[] players = {"Annette", "Bertrand",  
                    "Claude", "Danielle"};  
Game game = new Game("Bauernschnapsen", players);  
System.out.println( < missing expression > );
```

Which of the following can replace < missing expression > so that the code results in printing "Annette"?

- (A) game.getPlayer(0) — get Player → players.get(0-1) → I.O.B error
- (B) game.getPlayer(1)
- (C) game.players.get(0) players is private
- (D) game.players.get(1)
- (E) game.getPlayers().get(0) no getPlayers method in Game

#6 Which of the following can replace < missing code > in ChessGame's constructor so that the statement

```
System.out.println(new ChessGame("Deep Blue",  
                                "Kasparov"));
```

prints

Chess game [Deep Blue, Kasparov]

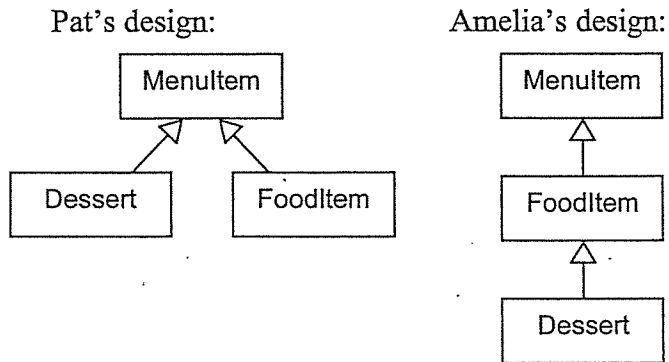
- I. super("Chess", white, black);
- II. super("Chess");
super.addPlayer(white);
super.addPlayer(black);
- III. String[] players = {black, white};
super("Chess", players);

Game has no constructor w/ three parameters

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

#7

Amelia and Pat are working on a programming project for a restaurant. They are considering a hierarchy of classes that includes, among other classes, MenuItem, FoodItem, and Dessert. Pat, who likes to eat his dessert first, is proposing to make both FoodItem and Dessert direct subclasses of MenuItem; Amelia's proposal is to make FoodItem a subclass of MenuItem and Dessert a subclass of FoodItem.



Which of the following is an advantage of Pat's design as compared to Amelia's?

- I. Both FoodItem and Dessert can reuse public methods of MenuItem.
 - II. Both FoodItem and Dessert type of object can be passed to a method that accepts a MenuItem as a parameter.
 - III. Pat's design better reflects the IS-A relationships between the three classes.
- (A) None of the three
(B) I only
(C) II only
(D) I and II only
(E) I, II, and III