

COMPUTER SCIENCE A**SECTION I****Time—1 hour and 30 minutes****Number of Questions—40****Percent of total exam grade—50%**

Directions: Determine the answer to each of the following questions or incomplete statements, using the available space for any necessary scratchwork. Then decide which is the best of the choices given and fill in the corresponding oval on the answer sheet. No credit will be given for anything written in the examination booklet. Do not spend too much time on any one problem.

Notes:

- Assume that the classes listed in the Quick Reference have been imported where appropriate.
- Assume that declarations of variables and methods appear within the context of an enclosing class.
- Assume that method calls that are not prefixed with an object or class name and are not shown within a complete class definition appear within the context of an enclosing class.
- Unless otherwise noted in the question, assume that parameters in the method calls are not `null` and that methods are called only when their preconditions are satisfied.

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

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

2. Which of the following expressions does **not** evaluate to 0.2?

- (A) $(1.0 * 2) / (1.0 * 10)$
 (B) $2.0 / 10$
 (C) $(\text{double}) 2 / 10$
 (D) $(\text{double})(2 / 10)$
 (E) `Math.sqrt(4) / Math.sqrt(100)`

3. Choose the code used to print the following:

`"Friends"`

- (A) `System.out.print("Friends");`
 (B) `System.out.print("//Friends//");`
 (C) `System.out.print("//Friends//");`
 (D) `System.out.print("\\"Friends\\");`
 (E) `System.out.print("\\\"Friends \\\"");`

GO ON TO THE NEXT PAGE.

4. 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

```

Questions 5–6 refer to the Constellation class below.

```

public class Constellation {
    private String name;
    private String month;
    private int northernLatitude;
    private int southernLatitude;

    Constellation(String n, String m)
    {
        name = n;
        month = m;
        northernLatitude = 0;
        southernLatitude = 0;
    }

    Constellation(String n, String m, int nLat, int sLat)
    {
        name = n;
        month = m;
        northernLatitude = nLat;
        southernLatitude = sLat;
    }

    public void chgMonth(String m)
    {
        String month = m;
    }
}

```

5. Using the Constellation class, which of the following will cause a compiler error?

- (A) Constellation c1 = new Constellation("Hercules", "July");
- (B) Constellation c2 = new Constellation("Pisces", "Nov", 90, 65);
- (C) Constellation c3 = new Constellation("Aquarius", "Oct", 65.0, 90.0);
- (D) Constellation c4 = new Constellation("Leo", "4", 0, 0);
- (E) Constellation c5 = new Constellation("Phoenix", "Nov", 32, 90);

6. A programmer has attempted to add three mutator methods to the Constellation class.

I. public void chgLatitude(String direction, int latitude)
{

```
    if (direction.toUpperCase().equals("N"))
        northernLatitude = latitude;
    else if (direction.toUpperCase().equals("S"))
        southernLatitude = latitude;
```

}

II. public void chgLatitude(int nLatitude, int sLatitude)
{

```
    northernLatitude = nLatitude;
    southernLatitude = sLatitude;
```

}

III. public void chgLatitude(double nLatitude, double sLatitude)
{

```
    northernLatitude = (int) nLatitude;
    southernLatitude = (int) sLatitude;
```

}

}

Which of the three will compile without a compiler error?

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

7. Determine the output of the following code.

```

int x = 10;
int y = 5;

if (x == 10)
{
    if (y <= 5)
        y++;
    else if (y < 4)
        x=3;
    else
        y+=6;
}
if (y > 5)
{
    if (x != 10)
    {
        x = 0;
        y = 0;
    }
    else
        x = -5;
}

```

- (A) $x = 0, y = 0$
- (B) $x = -5, y = 6$
- (C) $x = 10, y = 5$
- (D) $x = 3, y = 5$
- (E) None of the above

GO ON TO THE NEXT PAGE.

8. 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

9. Using the following variable declarations, determine which of the following would evaluate to true.

```

int temp = 90;
boolean cloudy = false;

```

- I. if (temp >= 90 && !cloudy)
- II. if (!(temp > 90 || cloudy))
- III. if (!(temp > 90 && !cloudy))

- (A) I only
- (B) II only
- (C) III only
- (D) Two of the above will evaluate to true.
- (E) All the above will evaluate to true.

10. Consider the following code:

```

1  String dog1 = new String("Poodle");
2  String dog2 = new String("Beagle");
3  dog1 = dog2;
4  String dog3 = new String("Beagle");
5
6  if (dog1 == dog2)
7      System.out.println("dog1 and dog2 are one and the same dog");
8  else
9      System.out.println("dog1 and dog2 are not the same dog");
10
11 if (dog1 == dog3)
12     System.out.println("dog1 and dog3 are one and the same dog");
13 else
14     System.out.println("dog1 and dog3 are not the same dog");
15
16 if (dog1.equals(dog3))
17     System.out.println("dog1 and dog3 are the same breed");
18 else
19     System.out.println("dog1 and dog3 are not the same breed");

```

Which of the following represents the output that will be produced by the code?

- (A) dog1 and dog2 are one and the same dog
dog1 and dog3 are one and the same dog
dog1 and dog3 are the same breed
- (B) dog1 and dog2 are one and the same dog
dog1 and dog3 are one and the same dog
dog1 and dog3 are not the same breed
- (C) dog1 and dog2 are one and the same dog
dog1 and dog3 are not the same dog
dog1 and dog3 are the same breed
- (D) dog1 and dog2 are one and the same dog
dog1 and dog3 are not the same dog
dog1 and dog3 are not the same breed
- (E) dog1 and dog2 are not the same dog
dog1 and dog3 are not the same dog
dog1 and dog3 are the same breed

11. 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)

12. Consider the following code excerpt:

```

9 int n = // some integer greater than zero
10 int count = 0;
11 int p = 0;
12 int q = 0;
13 for (p=1; p < n; p++)
14     for (q=1; q <= n; q++)
15         count++;

```

What will be the final value of count?

- (A) n^n
- (B) $n^2 - 1$
- (C) $(n - 1)^2$
- (D) $n(n - 1)$
- (E) n^2

13. Given the following code excerpt, determine the output.

```

1 int x = 0;
2 for (int j = 1; j < 4; j++)
3 {
4     if (x != 0 && j / x > 0)
5         System.out.print(j / x + " ");
6     else
7         System.out.print(j * x + " ");
8 }

```

- (A) 0 0 0
- (B) 0 0 0 0
- (C) 1 2 3
- (D) 1 0 2 0 3 0
- (E) ArithmeticException: Divide by Zero

GO ON TO THE NEXT PAGE.

14. Consider the following code:

```

1  String space = " ";
2  String symbol = "*";
3  int num = 5;
4  for (int i = 1; i <= num; i++)
5  {
6      System.out.print(symbol);
7  }
8  System.out.print("\n");
9  for (int i = 1; i <= num; i++)
10 {
11     for (int j = num - i; j > 0; j--)
12     {
13         System.out.print(space);
14     }
15     System.out.println(symbol);
16 }
17 for (int i = 1; i <= num; i++)
18 {
19     System.out.print(symbol);
20 }
```

Which of the following represents the output?

<p>(A) ***** ***** *** ** * *****</p>	<p>(D) ***** * * * * *****</p>
<p>(B) ***** **** *** ** * *****</p>	<p>(E) ***** * ** *** **** *****</p>
<p>(C) ***** * * * * *****</p>	

DO NOT TURN OVER UNTIL YOU ARE TOLD.

GO ON TO THE NEXT PAGE.

15. What will be printed as a result of the following code excerpt?

```
int sum = 0;
for (int i = 1; i < 2; i++)
    for (int j = 1; j <= 3; j++)
        for (int k = 1; k < 4; k++)
            sum += (i * j * k);

System.out.println(sum);
```

- (A) 18
 (B) 36
 (C) 45
 (D) 60
 (E) 108

16. Consider the following code:

```
1 int j = 0;
2 String s = "map ";
3 while ( j < s.length())
4 {
5     int k = s.length()
6     while ( k > j )
7     {
8         System.out.println(s.substring(j, k));
9         k--;
10    }
11    j++;
12 }
```

Which of the following represents the output?

(A) map ma m ap a	(D) m ma map a ap p
(B) map ma m ap a p	(E) p ap p map ma m
(C) map ap p ap p p	

GO ON TO THE NEXT PAGE.

17. A factorial is shown by an exclamation point(!) following a number. The factorial of 5 or $5!$ is calculated by $(5)(4)(3)(2)(1) = 120$.

Assuming n is an integer greater than 1, Choose the method that will return n!

I.	<pre>public static int f(int n) { int factorial = 1; for (int i = n; i > 0 ; i--) { factorial *= n; } return factorial;</pre>
II.	<pre>public static int f(int n) { int factorial = 1; int j = 1; while (j <= n) { factorial *= j; j++; } return factorial;</pre>
III.	<pre>public static int f(int n) { if (n==1) return n; return n * f(n-1);</pre>

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

18. Given the following code excerpt for the Tile Class:

```

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

```

What is the output after the following client code is executed?

```

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

```

- (A) Tile@5cccd43c2
- (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

GO ON TO THE NEXT PAGE.

19. What is the output after the following client code is executed?

```
Tile t2 = new Tile(101, "blue");
System.out.print(t2);
```

- (A) Tile@5cccd43c2
- (B) 101 blue 0.0 0.0 null 0.0
- (C) Type mismatch error
- (D) NullPointerException
- (E) There will be no output; the program will not compile.

20. 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;

21. Given the following code excerpt:

```
9  int[] nums = {11, 22, 33, 44, 55, 66};
10
11 for (int i = 0; i < nums.length; i++)
12     nums[nums[i] / 11] = nums[i];
```

Determine the final contents of **nums**.

- (A) 1, 2, 3, 4, 5, 6
- (B) 11, 11, 33, 33, 55, 55
- (C) 11, 11, 22, 33, 44, 55
- (D) 11, 22, 22, 33, 33, 55
- (E) 11, 22, 33, 44, 55, 66

GO ON TO THE NEXT PAGE.

22. Given the following code excerpt:

```
13 int[] arr1 = {1, 2, 3, 4, 5, 6};  
14 int[] arr2 = arr1;  
15 int last = arr1.length - 1;  
16  
17 for (int i = 0; i < arr1.length; i++)  
18     arr2[i] = arr1[last-i];  
19  
20 for (int i = 0; i < arr1.length; i++)  
21     System.out.print(arr1[i] + " ");  
22  
23 System.out.println(" ");  
24  
25 for (int i = 0; i < arr2.length; i++)  
26     System.out.print(arr2[i] + " ");
```

Determine the statement below that reflects the resulting output.

- (A) 1 2 3 4 5 6
 1 2 3 4 5 6
- (B) 1 2 3 4 5 6
 6 5 4 4 5 6
- (C) 6 5 4 3 2 1
 6 5 4 4 5 6
- (D) 6 5 4 4 5 6
 1 2 3 4 5 6
- (E) 6 5 4 4 5 6
 6 5 4 4 5 6

GO ON TO THE NEXT PAGE.

23. Given the following code excerpt:

```
27 int[] arr3 = {1, 2, 3, 4, 5, 6};  
28  
29 for (int element : arr3)  
30 {  
31     element *= 2;  
32     System.out.print(element + " ");  
33 }  
34 System.out.println(" ");  
35  
36 for (int element : arr3)  
37     System.out.print(element + " ");
```

Determine the statement below that reflects the resulting output.

- (A) 1 2 3 4 5 6
 1 2 3 4 5 6
- (B) 2 4 6 8 10 12
 1 2 3 4 5 6
- (C) 2 4 6 8 10 12
 2 4 6 8 10 12
- (D) A compiler error will occur.
- (E) A run-time exception will occur.

GO ON TO THE NEXT PAGE.

24. Given an array **numbers** containing a variety of integers and the following code excerpt:

```
38 int holdSmallest = Integer.MAX_VALUE;
39 int holdLargest = 0;
40 int a = 0;
41 int b = 0;
42 for (int i = 0; i < numbers.length; i++)
43 {
44     if (numbers[i] <= holdSmallest)
45     {
46         holdSmallest = numbers[i];
47         a = i;
48     }
49     if (numbers[i] >= holdLargest)
50     {
51         holdLargest = numbers[i];
52         b = i;
53     }
54 }
55 System.out.println(a + " " + b);
```

Determine the statement below that reflects the most successful outcome.

- (A) The code will print the smallest and largest values in the **numbers** array.
- (B) The code will print the locations of the smallest and largest values in the **numbers** array.
- (C) The code will print the locations of the smallest and largest non-negative values in the **numbers** array.
- (D) The code will print the location of the smallest value in the **numbers** array and the largest non-negative value in the **numbers** array.
- (E) The code will print the location of the smallest non-negative value in the **numbers** array and the largest value in the **numbers** array.

BLANK THIS SECTION OF THE PAGE

GO ON TO THE NEXT PAGE.

25. Choose the missing code below that will accurately find the average of the values in the sales array.

```

56 double avg = 0;
57 int i = 0;
58 int sum = 0;
59 for (int element : sales)
60
61 //Missing code
62
63

```

(A)	<pre> { sum += element; } double avg = (double) sum / sales.length; </pre>
(B)	<pre> { sum += sales[i]; } double avg = (double) sum / sales.length; </pre>
(C)	<pre> { sum += sales; } double avg = (double) sum / sales.length; </pre>
(D)	<pre> { sum += sales[element]; } double avg = (double) sum / sales.length; </pre>
(E)	<pre> { sum += element[sales]; } double avg = (double) sum / sales.length; </pre>

GO ON TO THE NEXT PAGE.

26. A programmer has written two different methods for a client program to swap the elements of one array with those of another array.

```

11  public static void swap1(int[] a1, int[] a2)
12  {
13      for (int i = 0; i < a1.length; i++)
14      {
15          int arrhold = a1[i];
16          a1[i] = a2[i];
17          a2[i] = arrhold;
18      }
19  }

20
21 public static void swap2(int[] a1, int[] a2) {
22     int [] arrhold= a1;
23     a1 = a2;
24     a2 = arrhold;
25 }
```

Which of the following statements best reflects the outcome of the two methods?

- (A) Both methods will swap the contents of the two arrays correctly in all cases.
- (B) swap1 will swap the contents of the two arrays correctly **only** if both arrays have the same number of elements, whereas swap2 will work correctly for all cases.
- (C) swap1 will only swap the contents of the two arrays correctly if both arrays have the same number of elements, whereas swap2 will **never** work correctly.
- (D) swap1 will only swap the contents of the two arrays correctly if both arrays have the same number of elements or a2 has more elements, whereas swap2 will work correctly for all cases.
- (E) Neither method will swap the contents of the two arrays correctly under any conditions.

27. Which code has declared and properly populated the given ArrayList?

I.	<code>ArrayList <String> alist1 = new ArrayList<String>(); alist1.add("4.5");</code>
II.	<code>ArrayList <Integer> alist2 = new ArrayList<Integer>(); alist2.add((int) 4.5);</code>
III.	<code>ArrayList <Double> alist3 ; alist3 = new ArrayList<Double>(); alist3.add(4.5);</code>

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

28. Given the following code excerpt:

```
ArrayList <Integer> alist1 = new ArrayList<Integer>();
int [] a1 = {2, 4, 6, 7, 8, 10, 11};
for (int a: a1) {
    alist1.add(a);
}
for (int i = 0; i < alist1.size(); i++) {
    if (alist1.get(i) % 2 == 0){
        alist1.remove(i);
    }
}
System.out.println(alist1);
```

Determine the output.

- (A) [4, 7, 10, 11]
- (B) [2, 4, 7, 10, 11]
- (C) [2, 7, 10, 11]
- (D) [7, 11]
- (E) An IndexOutOfBoundsException will occur

Questions 29–30 refer to the following code excerpt.

```
2   ArrayList <Integer> alist5 = new ArrayList<Integer>();
3   int [] a1 = {21, 6, 2, 8, 1};
4   for (int a: a1)
5   {
6       alist5.add(a);
7   }
8   for (int k = 0; k < alist5.size()-1; k++)
9   {
10      for (int i = 0; i < alist5.size()-2; i++)
11      {
12          if (alist5.get(i) > alist5.get(i + 1) )
13          {
14              int hold = alist5.remove(i);
15              alist5.add(i+1, hold );
16          }
17      }
18  }
19 System.out.println(alist5);
```

29. How many times will line 12 be executed?

- (A) 6 times
- (B) 12 times
- (C) 15 times
- (D) 16 times
- (E) 20 times

30. What will be the final output after the code executes?

- (A) [21, 8, 6, 2, 1]
- (B) [6, 21, 2, 8, 1]
- (C) [6, 2, 8, 21, 1]
- (D) [2, 6, 8, 21, 1]
- (E) [1, 2, 6, 8, 21]

GO ON TO THE NEXT PAGE.

31. Given `nums`—a rectangular, but not necessarily square, two-dimensional array of integers, choose the code to correctly print the array:

```

4 int [][] arr2d = {{1, 2, 3, 4}, {5, 6, 7, 8}};
5 String s= "";
6 for (int a = 0; a < arr2d[0].length; a++)
7 {
8     for (int b = 0; b < arr2d.length; b++)
9     {
10         s +=arr2d [b][a] + " ";
11     }
12     s += "\n";
13 }
14 System.out.print(s);

```

Determine the resulting output.

(A) 1 2 3 4
5 6 7 8

(B) 1 5 2 6
3 7 4 8

(C) 1 2
3 4
5 6
7 8

(D) 1 5
2 6
3 7
4 8

(E) 1
2
3
4
5
6
7
8

STOP. TURN OVER TO THE OTHER SIDE.

GO ON TO THE NEXT PAGE.

32. 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

GO ON TO THE NEXT PAGE.

Questions 33–34 refer to the Percussion and Xylophone class below.

```

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;
    }
}

```

33. Which of the following is the most appropriate replacement for **<missing code>** in the Xylophone constructor?

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

34. Assuming the above classes compile correctly, which of the following will not compile within a client program?

- (A) Xylophone [] xylophones = **new** Xylophone[5];
- (B) Percussion [] xylophones = **new** Xylophone[5];
- (C) Xylophone x1 = **new** Xylophone ("xylophone", 65, 32);
System.out.println(x1.getNumKeys());
- (D) Xylophone x1 = **new** Xylophone ("xylophone", 65, 32);
System.out.println(x1.numberofKeys());
- (E) Drums [] drums;

35. 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 weight of each object cannot be compared.

DEPARTMENT OF NO OF

GO ON TO THE NEXT PAGE.

Questions 36–37 refer to the following classes.

```

public class Dog {
    private int height;
    private String size;
    private String color;
    Dog (int iheight, int iweight, String icolor)
    {
        height = iheight;
        color = icolor;
        if (iweight >= 65)
            size = "large";
        else
            size = "medium";
    }
    public int getheight() {return height;}
    public String getSize() {return size;}
    public String getColor() {return color;}
    public String toString() {return "    color is: " + color;}
}

public class SportingDog extends Dog {
    private String purpose;
    SportingDog(int h, int w, String c)
    {
        super(h, w, c);
        purpose = "hunting";
    }
    public String getPurpose()
    {
        return purpose;
    }
}

public class Retriever extends SportingDog{
    private String type;
    Retriever(String itype, String icolor, int iweight)
    {
        super(24, iweight, icolor);
        type = itype;
    }
    public String toString() {return "    type: " + type + super.toString();}
}

```

36. Which of the following declarations will not compile?

- (A) Dog d1 = new SportingDog(30, 74, "Black");
- (B) Dog d2 = new Retriever("Labrador", "yellow", 75);
- (C) SportingDog d3 = new Retriever("Golden", "Red", 70);
- (D) SportingDog d4 = new Dog(25, 80, "Red");
- (E) Retriever d5 = new Retriever("Golden", "Blonde", 60);

DO NOT TURN THE PAGE OR GO ON.

GO ON TO THE NEXT PAGE.

37. What is the output after the execution of the following code in the client program:

```
Dog mason = new Retriever("Labrador", "chocolate", 85);
System.out.println(mason.toString());
```

- (A) type: Labrador
- (B) type: Labrador color is: chocolate purpose: hunting
- (C) color is: chocolate type: Labrador
- (D) type: Labrador purpose: hunting color is: chocolate
- (E) type: Labrador color is: chocolate

38. The following pow method was written to return b raised to the xth power where $x > 0$, but it does not work properly. Choose the changes to the method below to work properly.

```
1 public double pow(double b, int x)
2 {
3     if (x==0)
4         return 0;
5     else
6         return b + pow (b, x-1);
7 }
```

- (A) Change lines 3 and 4 to:

```
3     if (x==1)
4         return 1;
```

- (B) Change lines 3 and 4 to:

```
3     if (x==1)
4         return b;
```

- (C) Change line 6 to:

```
6     return b * mystery(b, x-1);
```

- (D) Both (A) and (C)
- (E) Both (B) and (C)

39. What is output given the following code excerpt?

```
System.out.println(f(8765));
public static int f(int n)
{
    if (n == 0)
        return 0;
    else
        return f(n/10) + n % 10;
}
```

- (A) 5678
- (B) 8765
- (C) 58
- (D) 26
- (E) A run-time error

GO ON TO THE NEXT PAGE.

40. 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

```

END OF SECTION I

**IF YOU FINISH BEFORE TIME IS CALLED,
YOU MAY CHECK YOUR WORK ON THIS SECTION.**

DO NOT GO ON TO SECTION II UNTIL YOU ARE TOLD TO DO SO.

COMPUTER SCIENCE A**SECTION II****Time—1 hour and 30 minutes****Number of Questions—4****Percent of Total Grade—50%****Directions:** SHOW ALL YOUR WORK. REMEMBER THAT PROGRAM SEGMENTS ARE TO BE WRITTEN IN JAVA™.**Notes:**

- Assume that the classes listed in the Java Quick Reference have been imported where appropriate.
- Unless otherwise noted in the question, assume that parameters in method calls are not null and that methods are called only when their preconditions are satisfied.
- In writing solutions for each question, you may use any of the accessible methods that are listed in classes defined in that question. Writing significant amounts of code that can be replaced by a call to one of these methods will not receive full credit.

FREE-RESPONSE QUESTIONS

- This question involves the implementation of a simulation rolling two dice. A client program will specify the number of rolls of the sample size and the number of faces on each of the two dice. A method will return the percentage of times the roll results in a double. Double in this case means when two dice match or have the same value (not a data type).

You will write two of the methods in this class.

```
public class DiceSimulation {
    /**
     * Sample size of simulation
     */
    private int numSampleSize;

    /**
     * Number of faces on the die
     */
    private int numFaces;

    /**
     * Constructs a DiceSimulation where sampleSize is the number of rolls to be simulated and
     * faces is the number of faces on the die (some die have many more or less than 6 faces)
     */
    public DiceSimulation(int numSamples, int faces) {
        numSampleSize = numSamples;
        numFaces = faces;
    }

    /**
     * Returns an integer from 1 to the number of faces to simulate a die roll */
    public int roll() {
        implemented in part (a)
    }

    /**
     * Simulates rolling two die with the number faces given, for the number of sample size
     * rolls. Returns the percentage of matches that were rolled
     * as an integer (eg. 0.50 would be 50)
     */
    public int runSimulation() {
        implemented in part (b)
    }
}
```

GO ON TO THE NEXT PAGE.

The following table contains sample code and the expected results.

Statements and Expressions	Value Returned / Comment
DiceSimulation s1 = new DiceSimulation(10, 6)	(no value returned) A DiceSimulation d1 is declared and instantiated.
s1.runSimulation()	10 rolls are simulated, only the percentage of matches is displayed. See further explanation below.
The 10 rolls might look like this (nothing is printed at this time) Die1: 3 Die2: 4 Die1: 1 Die2: 5 Die1: 2 Die2: 2 Die1: 3 Die2: 4 Die1: 6 Die2: 6 Die1: 3 Die2: 4 Die1: 3 Die2: 3 Die1: 6 Die2: 4 Die1: 3 Die2: 1 Die1: 5 Die2: 5	The percentage the method would return is 40.

- (a) Write the roll method to simulate the roll of one die.

Class information for this question

```
public class DiceSimulation
{
    private int numSampleSize;
    private int numFaces;

    public DiceSimulation (int numSamples, int faces)
    public int roll()
    public int runSimulation()
```

WRITE YOUR SOLUTION BELOW

```
/** Returns an integer from 1 to number of faces to simulate a die roll */
public int roll()
```

DO NOT GO ON TO PROBLEMS A UNTIL YOU ARE TOLD TO DO SO

GO ON TO THE NEXT PAGE.

- (b) Write the runSimulation method.

Class information for this question

```
public class DiceSimulation
private int numSampleSize;
private int numFaces;

public DiceSimulation (int numSamples, int faces)
public int roll()
public int runSimulation()
```

Requirement	Implementation	Test Example
01. defining class DiceSimulation to represent a dice simulation	public class DiceSimulation {	new DiceSimulation(100, 6)
02. defining constructor for DiceSimulation to accept sample size and number of faces	private int numSampleSize;	new DiceSimulation(100, 6)
03. defining roll() method to return a random integer between 1 and numFaces	private int numFaces;	roll()
04. defining runSimulation() method to accept sample size and number of faces and return average of rolls	public int runSimulation()	runSimulation(100, 6)

GO ON TO THE NEXT PAGE.

GO ON TO THE NEXT PAGE.

2. This question involves the implementation of a calorie counter system that is represented by the CalorieCount class. A CalorieCount object is created with 5 parameters:

- Daily Calories limit—the recommended number of calories per day
- Daily Calories intake—the number of calories a person has eaten in a day
- Grams of protein per day
- Grams of Carbohydrate per day
- Grams of Fat per day

The CalorieCount class provides a constructor and the following methods:

- addMeal—takes in calories, grams of protein, grams of carbs, and grams of fat from a meal and updates corresponding instance fields
- getProteinPercentage—returns the percent of protein in a given day ($4 * \text{grams protein} / \text{daily calorie intake}$)
- onTrack—returns true if the calorie intake does not exceed the daily calories limit, otherwise returns false

The following table contains a sample code and the expected results.

Statements and Expressions	Value Returned (blank if no value)	Comment
CalorieCount sunday = new CalorieCount(1500);		Creates an instance with a 1500 calorie limit.
sunday.addMeal(716, 38, 38, 45);		Adds 716 calories, 38 grams protein, 38 grams of carbs, 45 grams of fat to the appropriate instance fields
sunday.addMeal(230, 16, 8, 16);		Adds 230 calories, 16 grams protein, 8 grams of carbs, 16 grams of fat to the appropriate instance fields
sunday.addMeal(568, 38, 50, 24);		Adds 568 calories, 38 grams protein, 50 grams of carbs, 24 grams of fat to the appropriate instance fields
onTrack()	False	Returns true if calorie intake does not exceed calorie limit
getProteinPercentage()	.24	Multiplies grams of protein by 4 then divides by calorie intake

Write the entire CalorieCount class. Your implementation must meet all specifications and conform to all examples.

CalorieCount is a class that counts the total number of calories in a meal. It takes a list of strings representing the items in a meal and returns the total number of calories. The class has two main methods: `addFood` and `getCalories`. The `addFood` method adds a new food item to the meal. The `getCalories` method returns the total number of calories in the meal. The class also has a private field `meal` which stores the list of food items.

```
public class CalorieCount {
    private List<String> meal;

    public void addFood(String food) {
        meal.add(food);
    }

    public int getCalories() {
        int totalCalories = 0;
        for (String food : meal) {
            totalCalories += calculateCalories(food);
        }
        return totalCalories;
    }

    private int calculateCalories(String food) {
        // Implementation of calculateCalories goes here
    }
}
```

The `calculateCalories` method is a placeholder for the logic to calculate the calories for each food item. You will need to implement this method based on the requirements provided in the examples.

For example, if you call `addFood("apple")`, the total number of calories in the meal will be 50. If you call `addFood("banana")`, the total number of calories will be 100. If you call `getCalories()`, it will return 150.

You can assume that the input food items are valid strings and do not contain any punctuation or special characters. You can also assume that the meal list will not be empty.

Good luck!

GO ON TO THE NEXT PAGE.

GO ON TO THE NEXT PAGE.

3. This question involves the implementation of a Travel Planner system that is represented by the TravelPlan and Tour classes. A client will create Tour objects that will represent tours or activities of interest. Each Tour object is made up of an activity date, start time, end time, and name of the activity. The client will also create a TravelPlan object comprised of a destination and an arraylist of Tours. You will write three methods of the TravelPlan class.

A partial declaration of the Tour class is shown below.

```
public class Tour {
    private int actDate; // date is in mmddyyyy format
    private int startTime; // times are represented in military format
    private int endTime; // 1430 for 2:30 pm
    private String activity;

    /** Constructs a Tour
     * All instance fields are initialized
     */
    Tour(int actDate, int startTime, int endTime, String activity)
    {
        /* implementation not shown
     */
        public int getActDate() { return actDate; }
        public int getStartTime() { return startTime; }
        public int getEndTime() { return endTime; }
        public String getActivity() { return activity; }
    }
}
```

A partial declaration of the TravelPlan class is shown below.

```
import java.util.ArrayList;

public class TravelPlan {
    private String destination;
    private ArrayList <Tour> plans;

    /** Constructs a Tour
     * Instance fields: destination and plans are initialized
     */
    TravelPlan(String destination)
    {
        /* to be implemented in part (a) */
    }

    /** Returns true if the timeframe in t overlaps with another Tour in plans;
     * otherwise false
     */
    public boolean checkForConflicts(Tour t)
    {
        /* to be implemented in part (b) */
    }

    /** Must call checkForConflicts for full credit, if checkForConflicts returns false
     * (the timeframe does not overlap), adds t to plans. Returns true if
     * t was added, otherwise returns false
     */
    public boolean addTour(Tour t)
    {
        /* to be implemented in part (c) */
    }
}
```

The following table contains sample code and the expected results.

Statements and Expressions	Value Returned (blank if no value)	Comment
TravelPlan p1 = new TravelPlan("Capetown");		Creates an instance with a destination "CapeTown" and an empty arraylist of type Tour
Tours t1 = new Tours(12132020, 800, 1230, "Bungee jumping");		Create a Tour instance with date, start time, end time, and activity
Tours t2 = new Tours(12132020, 900, 1430, "Body surfing");		Create a Tour instance with date, start time, end time, and activity
p1.addtour(t1)	true	Checks for conflicts in plans, since there are none, adds the Tour object, returns true
p1.addtour(t2)	false	Checks for conflicts in plans, since there is a conflict, returns false
Tours t3 = new Tours(12132020, 1400, 1700, "Shark cage diving");		Create a Tour instance with date, start time, end time, and activity
p1.addtour(t3)	true	Checks for conflicts in plans, since there are none, adds the Tour object, returns true
Tours t4 = new Tours(12222020, 800, 1700, "Deep Sea Fishing");		Create a Tour instance with date, start time, end time, and activity
p1.addtour(t4)	true	Checks for conflicts in plans, since there are none, adds the Tour object, returns true

GO ON TO THE NEXT PAGE.

Section II

- (a) Write the TravelPlan constructor.

Class information for this question

```
public class Tour
private int actDate
private int startTime
private int endTime
private String activity

Tour(int actDate, int startTime, int endTime, String activity)
public int getActDate()
public int getStartTime()
public int getEndTime()
public String getActivity()

public class TravelPlan
private String destination;
private ArrayList <Tour> plans;

public TravelPlan(String destination)
public boolean checkForConflicts(Tour t)
public boolean addTour(Tour t)
```

GO ON TO THE NEXT PAGE.

GO ON TO THE NEXT PAGE.

Section II

(c) Write the TravelPlan addTour method.

GO ON TO THE NEXT PAGE

GO ON TO THE NEXT PAGE

4. This question involves the implementation of a class Seating Chart. A seating chart object will represent a two-dimensional String array. The number of rows and columns for the array will be sent as parameters, as well as a one-dimensional array of type Name. You may assume there will be enough rows and columns to accommodate all the entries from the array.

The declaration of the Name class is shown.

```
public class Name {
    private String lastName;
    private String firstName;

    Name(String lName, String fName){<implementation not shown>}
    public String getLastname() {return lastName;}
    public String getFirstName() {return firstName;}
}
```

A partial declaration of the SeatingChart class is shown below.

```
public class SeatingChart {
    private String [][] chart;

    /** Constructs a SeatingChart having r rows and c columns. All elements contained in the
     * names array should be placed randomly in the chart array using the format: last
     * Name, firstName
     * (e.g. Johlie, Angelina). Any locations not used in the chart should be
     * initialized to the empty string.
    */
    SeatingChart(Name[] names, int rows, int cols) {
        /* to be implemented in part (a) */
    }

    /** Returns a string containing all elements of the chart array in row-major order.
     * The method should return
     * a string containing all the elements in the chart array. The method
     * padWithSpaces should be called on each
     * element of chart before it is added to the string to ensure each name will be
     * printed with the same
     * length. Each row of the chart should be separated by a line break.
    */
    public String toString() {
        /* to be implemented in part (b) */
    }

    /** Pads a string with spaces to ensure each string is exactly 35 characters long. */

    private String padWithSpaces(String s) {
        String str = s;
        for (int a = s.length(); a < 35; a++) {
            str += " ";
        }
        return str;
    }
}
```

GO ON TO THE NEXT PAGE.

The following table contains sample code and the expected results.

Statements and Expressions	Value Returned / Comment	
SeatingChart msJones = new SeatingChart(theNames, 4, 3);	(no value returned) A two dimensional array is initialized with 4 rows and 3 columns. Every element in theNames is placed randomly in chart in the following format: lastname, firstname e.g., Washington, George. Empty string is placed in any unused locations.	
System.out.println(msJones.toString());	Prints the names in chart in row-major order. See example below:	
Miller, Minnie	Fitzgerald, Fred	Dade, Ali
Indigo, Inde	Banner, Boris	Lane, Lois
Titon, Tim	Robillard, Robbie	
Georgian, Greg		
Brne, Jane		

DO NOT WRITE ANYTHING ON THIS PAGE.

GO ON TO THE NEXT PAGE.

- (a) Write the SeatingChart constructor.

Class information for this question

```
public class Name
    private String lastName;
    private String firstName;

    Name(String lName, String fName)
    public String getLastName() {return lastName;}
    public String getFirstName() {return firstName;}

public class SeatingChart
    private String [][] chart;

    SeatingChart(Name[] names, int rows, int cols)
    public String toString()
    private String padWithSpaces(String s)
```

GO ON TO THE NEXT PAGE.

Section II

- (b) Write the `SeatingChart.toString()` method.

STOP

END OF EXAM
