Beijing National Day School
Department of Mathematics & Computer Science

AP Computer Science A

Semester 1 Exam

Location: Library, 6th Floor, Aspiration Building
Date: Monday, January 15th, 2018
Start Time: 8:00AM
End Time: 10:00AM
NO CALCULATORS PERMITTED

Exam Record

Multiple Choice _____ / 18 pts

Free Response _____ / 18 pts

Total: _____ / 36 pts

Grade: _____

English Name: _____

Pinyin Name: _____

Mr. Alwin Tareen, January 2018

Section I: Multiple Choice (18 points)

- Number of questions: 18. Percent of total grade: 50%. Suggested time: 60 minutes.

- Determine the answer to each of the following questions, using the available space for any necessary scratchwork.

- Decide which is the best of the choices given, and select the correct answer by placing an "**X**" in the corresponding box.

- Assume that the classes in the **Java Quick Reference** have been imported where needed.

- Assume that variables and methods are declared within the context of an enclosing class.

(1^{pt}) **1.** Consider the following code segment:

```java
for (int j = 0; j < 20; j = j+2)
{
    if (j % 3 == 1)
    {
        System.out.println(j + " ");
    }
}
```

1 pt

What will be printed as a result of executing this code segment?

☐ 0 2 4 6 8 10 12 14 16 18
☐ 4 16
☐ 0 6 12 18
☐ 1 4 7 10 13 16 19
☐ 4 10 16

(1^{pt}) **2.** Consider the following code segment:

```java
int var1 = 0;
int var2 = 2;

while ((var2 != 0) && ((var1 / var2) >= 0))
{
    var1 = var1 + 1;
    var2 = var2 - 1;
}
```

1 pt

Which of the following represents the contents of `var1` and `var2` as a result of executing this code segment?

☐ var1 = 0, var2 = 2
☐ var1 = 1, var2 = 1
☐ var1 = 3, var2 = -1
☐ var1 = 2, var2 = 0
☐ The loop won't finish executing, because of a division by zero.

(1pt)   **3.** Consider the following method:

```
public static double mystery(int[] arr)
{
    double result = 0;
    for (int i = 0; i < arr.length; i++)
    {
        result = result + arr[i];
    }
    return result / arr.length;
}
```

1 pt

What will be printed as a result of executing the following code segment?

```
int[] nums = {10, 30, 30, 60};
System.out.println(mystery(nums));
```

☐ 17.5
☐ 30.0
☐ 130
☐ 32
☐ 32.5

(1pt)   **4.** Consider the following code segment:

```
String myString = "H";
int index = 0;
while (index < 4)
{
    for (int i = 0; i < index; i++)
    {
        myString = "A" + myString + "A";
    }
    index++;
}
```

1 pt

What will be the value of `myString` after executing this code segment?

☐ "H"
☐ "AHA"
☐ "AAAHAAA"
☐ "AAAAHAAAA"
☐ "AAAAAAHAAAAAA"

(1pt)   **5.** Consider the following code segment:

```
int[] arr = {-10, -5, 1, 4, 8, 30};
for (int i = arr.length / 2; i < arr.length; i++)
{
    arr[i] = arr[i] * 2;
}
```

1 pt

Which of the following represents the contents of `arr` as a result of executing this code segment?

☐ {-20, -10, 2, 8, 16, 60}
☐ {-20, -10, 2, 4, 8, 30}
☐ {-10, -5, 1, 8, 16, 60}
☐ {-10, -5, 1, 4, 8, 30}
☐ {-10, -5, 1, 12, 24, 90}

(1pt)   **6.** Consider the following code segment:

```
int[] myArray = {2, 3, 4, 1, 7, 6, 8};
int index = 0;
while (myArray[index] < 7)
{
    myArray[index] += 3;
    index++;
}
```

1 pt

What values are stored in `myArray` after executing this code segment?

☐ {2, 3, 4, 1, 7, 6, 8}
☐ {2, 3, 4, 1, 7, 9, 11}
☐ {5, 6, 7, 4, 7, 9, 8}
☐ {5, 6, 7, 4, 7, 6, 8}
☐ {5, 6, 7, 4, 0, 9, 11}

(1pt)   **7.** Consider the following code segment:

```
int[][] matrix = {{1,1,2,2}, {1,2,2,4}, {1,2,3,4}, {1,4,1,2}};
int sum = 0;
int col = matrix[0].length - 2;
for (int row = 0; row < 4; row++)
{
    sum = sum + matrix[row][col];
}
```

1 pt

Which of the following represents the contents of `sum` as a result of executing this code segment?

☐ 4
☐ 8
☐ 9
☐ 12
☐ 10

(1ᵖᵗ)   **8.** Consider the following code segment:

```
ArrayList<String> sports = new ArrayList<String>();
sports.add("Archery")
sports.add("Lacrosse")
sports.add("Soccer")
sports.set(0, "Diving")
sports.add(0, "Swimming")
System.out.println(sports);
```

1 pt

What will be printed as a result of executing this code segment?

☐ ["Swimming", "Diving", "Lacrosse", "Soccer"]

☐ ["Swimming", "Diving", "Archery", "Lacrosse", "Soccer"]

☐ ["Swimming", "Lacrosse", "Soccer"]

☐ ["Diving", "Lacrosse", "Soccer", "Swimming"]

☐ ["Diving", "Lacrosse", "Soccer"]

(1ᵖᵗ)   **9.** Consider the following method:

```
public static String scramble(String word, int howFar)
{
  return word.substring(howFar+1,word.length()) + word.substring(0,howFar);
}
```

1 pt

What value is returned as a result of the call `scramble("compiler", 3)`?

☐ "compiler"

☐ "pilercom"

☐ "ilercom"

☐ "ilercomp"

☐ No value is returned because an `IndexOutOfBoundsException` will be thrown.

(1ᵖᵗ)   **10.** Consider the following method:

```
public String mystery(String input)
{
    String output = "";
    for (int k = 1; k < input.length(); k = k+2)
    {
        output += input.substring(k, k+1);
    }
    return output;
}
```

1 pt

What value is returned as a result of the call `mystery("computer")`?

☐ "computer"

☐ "cmue"

☐ "optr"

☐ "ompute"

☐ Nothing is returned because an `IndexOutOfBoundsException` is thrown.

(1pt) **11.** Consider the following partial class declaration.

```java
public class Park
{
    private String name;
    private boolean playground;
    private int acres;

    public Park(String myName, boolean myPlayground, int myAcres)
    {
        name = myName;
        playground = myPlayground;
        acres = myAcres;
    }

    public String getName()
    {
        return name;
    }

    public boolean hasPlayground()
    {
        return playground;
    }

    public int getAcres()
    {
        return acres;
    }

    /* Additional implementation not shown */
}
```

1 pt

Assume that the following declaration has been made in the `main` method of another class.

```java
Park park = new Park("Central", true, 300);
```

Which of the following statements compiles without error?

☐ `int num = park.acres;`
☐ `String name = central.getName();`
☐ `boolean play = park.hasPlayground();`
☐ `int num = park.getAcres(acres);`
☐ `park.hasPlayground = true;`

(1pt) **12.** Consider the following code segment:

```
for (int k = 1; k <= 100; k++)
{
    if (k % 4 == 0)
    {
        System.out.println(k);
    }
}
```

1 pt

Which of the following code segments will produce the same output as the code segment above?

☐ ```
for (int k = 1; k <= 25; k++)
{System.out.println(k);}
```

☐ ```
for (int k = 1; k <= 100; k = k+4)
{System.out.println(k);}
```

☐ ```
for (int k = 1; k <= 100; k++)
{System.out.println(k % 4);}
```

☐ ```
for (int k = 4; k <= 25; k = 4*k)
{System.out.println(k);}
```

☐ ```
for (int k = 4; k <= 100; k = k+4)
{System.out.println(k);}
```

(1pt) **13.** Assume that the array `arr` has been defined and initialized as follows:

```
int[] arr = /* initial values for the array */ ;
```

1 pt

Which of the following will correctly print all of the odd integers contained in `arr` but none of the even integers contained in `arr`?

☐ ```
for (int x : arr)
{if (x % 2 == 1)
System.out.println(x);}
```

☐ ```
for (int k = 1; k < arr.length; k++)
{if (arr[k] % 2 == 1)
System.out.println(arr[k]);}
```

☐ ```
for (int x :  arr)
{if (x % 2 == 1)
System.out.println(arr[x]);}
```

☐ ```
for (int k = 0; k < arr.length; k++)
{if (arr[k] % 2 == 1)
System.out.println(k);}
```

☐ ```
for (int x :  arr)
{if (arr[x] % 2 == 1)
System.out.println(arr[x]);}
```

(1pt)   **14.** Consider the following code segment:

```
ArrayList<String> students = new ArrayList<String>();
students.add("Alex");
students.add("Bob");
students.add("Carl");

for (int k = 0; k < students.size(); k++)
{
    System.out.print(students.set(k, "Alex") + " ");
}
System.out.println();

for (String name : students)
{
    System.out.print(name + " ");
}
```

1 pt

What will be printed as a result of executing this code segment?

☐ Alex Alex Alex
  Alex Alex Alex

☐ Alex Alex Alex
  Alex Bob Carl

☐ Alex Bob Carl
  Alex Alex Alex

☐ Alex Bob Carl
  Alex Bob Carl

☐ Nothing will be printed, because the first `print` statement will cause a runtime error to be thrown.

(1pt)   **15.** A pair of number cubes is used in a game of chance. Each number cube has six sides, numbered from 1 to 6, inclusive. There is an equal probability for each of the numbers to appear on the top side(indicating the cube's value) when the number cube is rolled. The following incomplete statement appears in a program that computes the sum of the values produced by rolling two number cubes.

1 pt

```
int sum = /* missing code */ ;
```

Which of the following replacements for `/* missing code */` would best simulate the value produced as a result of rolling two number cubes?

☐ `2 * (int) (Math.random() * 6)`
☐ `2 * (int) (Math.random() * 7)`
☐ `(int) (Math.random() * 6) + (int) (Math.random() * 6)`
☐ `(int) (Math.random() * 13)`
☐ `2 + (int) (Math.random() * 6) + (int) (Math.random() * 6)`

Questions 16, 17 and 18 refer to the `Time` class declared below:

```
1   public class Time
2   {
3       private int myHrs;
4       private int myMins;
5       private int mySecs;
6
7       public Time()
8       {
9           /* implementation not shown */
10      }
11
12      public Time(int h, int m, int s)
13      {
14          /* implementation not shown */
15      }
16
17      // Resets time to myHrs = h, myMins = m, mySecs = s.
18      public void resetTime(int h, int m, int s)
19      {
20          /* implementation not shown */
21      }
22
23      // Advances time by one second.
24      public void increment()
25      {
26          /* implementation not shown */
27      }
28
29      // Returns true if this time equals t, false otherwise.
30      public boolean equals(Time t)
31      {
32          /* implementation not shown */
33      }
34
35      // Returns true if this time is earlier than t, false otherwise.
36      public boolean lessThan(Time t)
37      {
38          /* implementation not shown */
39      }
40
41      // Returns time as a String in the form hrs:mins:secs.
42      public String toString()
43      {
44          /* implementation not shown */
45      }
46  }
```

(1^{pt}) **16.** Which of the following is a **false** statement about the methods?

☐ equals, lessThan, and toString are all accessor methods.
☐ increment is a mutator method.
☐ Time() is the default constructor.
☐ The Time class has three constructors.
☐ There are no static methods in this class.

1 pt

(1^{pt}) **17.** Which of the following represents correct **implementation code** for the constructor with parameters?

1 pt

☐ myHrs = 0;
   myMins = 0;
   mySecs = 0;

☐ myHrs = h;
   myMins = m;
   mySecs = s;

☐ resetTime(myHrs, myMins, mySecs);

☐ h = myHrs;
   m = myMins;
   s = mySecs;

☐ Time = new Time(h, m, s);

(1^{pt}) **18.** A **client class** called TimeTest has a display method that writes the time represented by its parameter:

1 pt

```
// Outputs time t in the form hrs:mins:secs.
public void display(Time t)
{
    /* method body */
}
```

Which of the following are correct replacements for /* method body */?

   I Time clock = new Time(h, m, s);
      System.out.println(clock);

   II System.out.println(t.myHrs + ":" + t.myMins + ":" + t.mySecs);

   III System.out.println(t);

☐ I only
☐ II only
☐ III only
☐ II and III only
☐ I, II and III

Section II: Free Response (18 points)

- Number of questions: 2. Percent of total grade: 50%. Suggested time: 60 minutes.

- **SHOW ALL YOUR WORK. REMEMBER THAT PROGRAM SEGMENTS ARE TO BE WRITTEN IN JAVA.**

- 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 may not receive full credit.

CookieOrder question(2010 AP Computer Science A Free Response)

(9^{pts}) **1.** An organization raises money by selling boxes of cookies. A cookie order specifies the variety of cookie and the number of boxes ordered. The declaration of the `CookieOrder` class is shown below.

9 pts

```java
public class CookieOrder
{
    /** Constructs a new CookieOrder object.
     */
    public CookieOrder(String variety, int numBoxes)
    { /* implementation not shown */ }

    /** @return the variety of cookie being ordered.
     */
    public String getVariety()
    { /* implementation not shown */ }

    /** @return the number of boxes being ordered.
     */
    public int getNumBoxes()
    { /* implementation not shown */ }

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

The MasterOrder class maintains a list of the cookies to be purchased. The declaration of the MasterOrder class is shown below.

```java
public class MasterOrder
{
    /** The list of all cookie orders.
     */
    private ArrayList<CookieOrder> orders;

    /** Constructs a new MasterOrder object.
     */
    public MasterOrder()
    { orders = new ArrayList<CookieOrder>(); }

    /** Adds theOrder to the master order.
     * @param theOrder the cookie order to add to the master order.
     */
    public void addOrder(CookieOrder theOrder)
    { orders.add(theOrder); }

    /** @return the sum of the number of boxes of all of the cookie orders.
     */
    public int getTotalBoxes()
    { /* to be implemented in part (a) */ }

    /** Removes all cookie orders from the master order that have the same variety of
     * cookie as cookieVar and returns the total number of boxes that were removed.
     * @param cookieVar the variety of cookies to remove from the master order.
     * @return the total number of boxes of cookieVar in the cookie orders removed.
     */
    public int removeVariety(String cookieVar)
    { /* to be implemented in part (b) */ }

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

(a) (3 pts) The `getTotalBoxes` method computes and returns the sum of the number of boxes of all cookie orders. If there are no cookie orders in the master order, the method returns `0`.

Complete method `getTotalBoxes` below.

```
/** @return the sum of the number of boxes of all of the cookie orders.
 */
public int getTotalBoxes()
```

(b) The `removeVariety` method updates the master order by removing all of the cookie orders in which the variety of cookie matches the parameter `cookieVar`. The master order may contain zero or more cookie orders with the same variety as `cookieVar`. The method returns the total number of boxes removed from the master order.

For example, consider the following code segment:

```
1  MasterOrder goodies = new MasterOrder();
2  goodies.addOrder(new CookieOrder("Chocolate Chip", 1));
3  goodies.addOrder(new CookieOrder("Shortbread", 5));
4  goodies.addOrder(new CookieOrder("Macaroon", 2));
5  goodies.addOrder(new CookieOrder("Chocolate Chip", 3));
```

After the code segment has executed, the contents of the master order are as shown in the following table.

"Chocolate Chip"	"Shortbread"	"Macaroon"	"Chocolate Chip"
1	5	2	3

The method call `goodies.removeVariety("Chocolate Chip")` returns 4 because there were two Chocolate Chip cookie orders totalling 4 boxes. The master order is modified as shown below.

"Shortbread"	"Macaroon"
5	2

The method call `goodies.removeVariety("Brownie")` returns 0 and does **not** change the master order.

Write your solution on the next page.

(b) (6 pts) Complete method `removeVariety` below.

```
/** Removes all cookie orders from the master order that have the same variety of
 * cookie as cookieVar and returns the total number of boxes that were removed.
 * @param cookieVar the variety of cookies to remove from the master order.
 * @return the total number of boxes of cookieVar in the cookie orders removed.
 */
public int removeVariety(String cookieVar)
```

GrayImage question(2012 AP Computer Science A Free Response)

(9^{pts})   **2.** A grayscale image is represented by a 2-dimensional rectangular array of pixels(picture elements). A pixel is an integer value that represents a shade of gray. In this question, pixel values can be in the range from 0 through 255, inclusive. A black pixel is represented by 0, and a white pixel is represented by 255.

9 pts

The declaration of the GrayImage class is shown below. You will write two unrelated methods of the GrayImage class.

```
1  public class GrayImage
2  {
3      public static final int BLACK = 0;
4      public static final int WHITE = 255;
5
6      /** The 2−dimensional representation of this image. Guaranteed not to be null.
7       * All values in the array are within the range [BLACK, WHITE], inclusive.
8       */
9      private int[][] pixelValues;
10
11     /** @return the total number of white pixels in this image.
12      * Postcondition: this image has not been changed.
13      */
14     public int countWhitePixels()
15     { /* to be implemented in part (a) */ }
16
17     /** Processes this image in row−major order and decreases the value of each pixel at
18      * position(row, col) by the value of the pixel at position(row+2, col+2) if it exists.
19      * Resulting values that would be less than BLACK are replaced by BLACK.
20      * Pixels for which there is no pixel at position(row+2, col+2) are unchanged.
21      */
22     public void processImage()
23     { /* to be implemented in part (b) */ }
24 }
```

(a) Write the method countWhitePixels that returns the number of pixels in the image that contain the value WHITE. For example, assume that pixelValues contains the following image:

	0	1	2	3	4
0	**255**	184	178	84	129
1	84	**255**	**255**	130	84
2	78	**255**	0	0	78
3	84	130	**255**	130	84

A call to the countWhitePixels method would return 5 because there are 5 entries(shown in boldface) that have the value WHITE.

(a) (4 pts) Complete method `countWhitePixels` below.

```
/** @return the total number of white pixels in this image.
 * Postcondition: this image has not been changed.
 */
public int countWhitePixels()
```

(b) Write the method `processImage` that modifies the image by changing the values in the instance variable `pixelValues` according to the following description. The pixels in the image are processed one at a time in **row-major order**.

Row-major order processes the first row in the array from left-to-right, and then processes the second row from left-to-right, continuing until all rows are processed from left-to-right. The first index of `pixelValues` represents the row number, and the second index represents the column number.

The pixel value at position(row, col) is decreased by the value at position (row+2, col+2) if such a position exists. If the result of the subtraction is less than the value `BLACK`, the pixel is assigned the value of `BLACK`. The values of the pixels for which there is no pixel at position(row+2, col+2) remain unchanged. You may assume that all the original values in the array are within the range `[BLACK, WHITE]`, inclusive.

The following diagram shows the contents of the instance variable `pixelValues` before and after a call to `processImage`. The values shown in boldface represent the pixels that could be modified in a grayscale image with 4 rows and 5 columns.

<div>

Before Call to processImage

	0	1	2	3	4
0	**221**	**184**	**178**	84	135
1	**84**	**255**	**255**	130	84
2	78	255	0	0	78
3	84	130	255	130	84

After Call to processImage

	0	1	2	3	4
0	**221**	**184**	**100**	84	135
1	**0**	**125**	**171**	130	84
2	78	255	0	0	78
3	84	130	255	130	84

</div>

Information repeated from the beginning of the question:

public class GrayImage

public static final int BLACK = 0
public static final int WHITE = 255
private int[ ][ ] pixelValues
public int countWhitePixels()
public void processImage()

Write your solution on the next page.

(b) (5 pts) Complete method `processImage` below.

```
/** Processes this image in row−major order and decreases the value of each pixel at
 * position(row, col) by the value of the pixel at position(row+2, col+2) if it exists.
 * Resulting values that would be less than BLACK are replaced by BLACK.
 * Pixels for which there is no pixel at position(row+2, col+2) are unchanged.
 */
public void processImage()
```

This page is left intentionally blank.

This page is left intentionally blank.