Fall 2013 **Final Exam** COSC 117

Name: _____

Write all of your responses on these exam pages. If you need more space please use the backs. Make sure that you show all of your work, answers without supporting work will receive no credit.

1 Definitions & Short Answer (5 Points Each)

1. Explain the difference between high-level languages and machine language.

2. What is the difference between a compiler and an interpreter? Also, discuss Java's method.

3. Java is a "platform-independent language." What is a *platform*, what does *platform-independent* mean, and how does Java attain its platform independence?

4. What are the three types of programming errors? Briefly describe each of them.

5. What is the scope of a method/function parameter?

6. What is method/function overloading? How is it done in Java?

7. What is the difference between a Java applet and a Java application? Specifically, what runs each of them and what is the main coding difference between the two?

2 Program Traces (15 Points Each)

1. Write the output of the following program for each of the three inputs.

```
1  import java.util.Scanner;
2
3  public class FinalTrace1 {
4
5      public static void main(String[] args) {
6          Scanner keyboard = new Scanner(System.in);
7          System.out.print("Input a b c: ");
8          int a = keyboard.nextInt();
9          int b = keyboard.nextInt();
10         int c = keyboard.nextInt();
11
12         System.out.printf("%d  %d  %d \n", a, b, c);
13         System.out.println(a/b);
14         System.out.println(c--);
15         System.out.println(++a);
16         b *= b;
17         System.out.printf("%d  %d  %d \n", a, b, c);
18
19         if ((a + b + c) % 2 == 0)
20             System.out.println((a + b + c)/2);
21
22         c = a++ + --b;
23         System.out.printf("%d  %d  %d \n", a, b, c);
24     }
25 }
```

Program Output

```
Input a b c: 7 3 6
```

Program Output

```
Input a b c: 3 5 7
```

Program Output

```
Input a b c: 4 0 2
```

2. Write the output of the following program for the given input. When you write the output put in a space marker for all spaces, including any beginning or trailing spaces. For example, the input below has no beginning or trailing spaces so it would be written as `This␣is␣a␣test.` on the other hand if one of the outputs is the word is with a single beginning and single trailing space you should write it as `␣is␣`.

```
1  import java.util.Scanner;
2
3  public class FinalTrace2 {
4
5      public static void main(String[] args) {
6          Scanner keyboard = new Scanner(System.in);
7          System.out.print("Input: ");
8          String s = keyboard.nextLine();
9
10         String s2 = s.substring(s.length()/4, s.length() - s.length()/4);
11         System.out.println(s2);
12
13         int pos1 = s.indexOf("is");
14         int pos2 = s.indexOf("is", pos1 + 1);
15         System.out.println(pos1 + " " + pos2);
16         String s3 = s.substring(pos2);
17         System.out.println(s3);
18
19         pos1 = s.indexOf("is ");
20         pos2 = s.indexOf(" is");
21         System.out.println(pos1 + " " + pos2);
22         s3 = s.substring(pos1, pos2);
23         System.out.println(s3);
24     }
25 }
```

Program Output

```
Input: This is a test.
```

3. Functions

4. Array 1d

5. Array 2d

3 Coding (20 Points Each)

1. Write a program that will take the number of trials as input from the user, then for each trial the program will roll three die and count the number of rolls that exceed 12 and the number of rolls that are less than 5. Have the program output the number of each type of roll along with the probability of each type of roll. Do the rolling of a single die in a method called roll.

2. Write four methods for the following main program. The methods are `stringReverse`, `removeSpaces`, `isPalindrome`, and `printPalCone`. Fron the code of the main program and the three test runs below you should be able to determine the parameter lists and return types that are needed as well as what each method does. Please write the code on the next page.

Main Program

```
public static void main(String[] args) {
    Scanner keyboard = new Scanner(System.in);
    System.out.print("Input a string: ");
    String str = keyboard.nextLine();

    String strns = removeSpaces(str);
    System.out.println("str: " + str);
    System.out.println("str no spaces: " + strns);
    String rstr = stringReverse(strns);
    System.out.println("rstr: " + rstr);
    boolean pal = isPalindrome(strns);
    System.out.println("str is a palindrome: " + pal);

    if (pal)
        printPalCone(strns);

    System.out.println("str: " + str);
}
```

Program Run

```
Input a string: ab cd  e dcb a
str: ab cd  e dcb a
str no spaces: abcdedcba
rstr: abcdedcba
str is a palindrome: true
ABCDEDCBA
 BCDEDCB
  CDEDC
   DED
    E
str: ab cd  e dcb a
```

Program Run

```
Input a string: A Toyota
str: A Toyota
str no spaces: AToyota
rstr: atoyoTA
str is a palindrome: true
ATOYOTA
 TOYOT
  OYO
   Y
str: A Toyota
```

Program Run

```
Input a string: Not a Palindrome
str: Not a Palindrome
str no spaces: NotaPalindrome
rstr: emordnilaPatoN
str is a palindrome: false
str: Not a Palindrome
```

Code for `stringReverse`, `removeSpaces`, `isPalindrome`, **and** `printPalCone`

3. Write either the Bubble Sort, Insertion Sort or Selection Sort (just do one of them) in a method that takes in one parameter only, a one-dimensional integer array named `A`.

4. Write a ChristmasList class that stores a person's name in the variable Name, an array of 5 strings called WishList and an array of 5 strings called SantasComments. The class should have five methods, each of them is discussed below. There is also a sample main program and its output that uses this class. Please write the code on the page after the example run.

   (a) `public ChristmasList(String n)`

   The constructor should store the input string in the Name variable and set all of the entries in both string lists to the empty string.

   (b) `public void addToWishlist(int i, String item)`

   This method should add the item into position i of the WishList. Note that the user (you) is going to think the list numbers go from 1 to 5 and not 0 to 4. If the user inputs a position outside the 1 to 5 range the program should print out an error message, like in the example below. If the user inputs a position that is already filled, the program will simply overwrite the old item. Look at the example code and run below.

   (c) `public void addToSantasComments(int i, String comment)`

   This method should add the comment into position i of SantasComments. Note that the user (Santa) is going to think the list numbers go from 1 to 5 and not 0 to 4. If the user inputs a position outside the 1 to 5 range the program should print out an error message, like in the example below. If the user inputs a position that is already filled, the program will simply overwrite the old comment. Look at the example code and run below.

   (d) `public void printChristmasList()`

   This method prints out the person's name, the number of items in their list and the list along with any of Santa's comments. For the number of items the method must call the numberItems method below. The list should only print out the items that have been input. So if the person has input items in positions 1, 3, and 5 then the printed list should only list items 1, 3, and 5 and skip items 2 and 4. Santa's comments for an item should only print out if the user has input an item in that position. That is, if Santa puts in a comment in position 2 and the person does not have an item in position 2 then Santa's comment will not be printed. Also, if the person has an item in a position but there is no comment then only the item is printed. If Santa does have a comment for an item, his comment should print out on the same line as the item with three dashes between the item and the comment. Look at the example code and run below.

   (e) `public int numberItems()`

   This method returns the number of items in the person's Christmas list. This should only count the items that have been input, for example, if the person has input items in positions 1, 3, and 5 then the number of items is 3 and not 5.

Main Program

```
public class ChristmasListMain {

    public static void main(String[] args) {
        ChristmasList mylist = new ChristmasList("Don Spickler");
        mylist.printChristmasList();
        mylist.addToWishlist(1, "New Car");
        mylist.addToSantasComments(1, "Due to extensive budget cuts this will not be possible.");
        mylist.printChristmasList();
        mylist.addToWishlist(5, "iPad");
        mylist.addToWishlist(3, "Sabbatical");
        mylist.printChristmasList();
        mylist.addToSantasComments(5, "Get a real tablet, Android.");
        mylist.addToSantasComments(2, "Nothing here????");
        mylist.printChristmasList();
        mylist.addToSantasComments(3, "You got it, Fall 2014 and Spring 2015!!!");
        mylist.printChristmasList();
        mylist.addToWishlist(6, "Puppy");
        mylist.addToSantasComments(6, "Error 404");
    }
}
```

Program Run

```
Christmas List for Don Spickler
Number of Christmas List items is 0

Christmas List for Don Spickler
Number of Christmas List items is 1
List
New Car --- Due to extensive budget cuts this will not be possible.

Christmas List for Don Spickler
Number of Christmas List items is 3
List
New Car --- Due to extensive budget cuts this will not be possible.
Sabbatical
iPad

Christmas List for Don Spickler
Number of Christmas List items is 3
List
New Car --- Due to extensive budget cuts this will not be possible.
Sabbatical
iPad --- Get a real tablet, Android.

Christmas List for Don Spickler
Number of Christmas List items is 3
List
New Car --- Due to extensive budget cuts this will not be possible.
Sabbatical --- You got it, Fall 2014 and Spring 2015!!!
iPad --- Get a real tablet, Android.

Cannot add to list, index is out of range.
Cannot add comment, index is out of range.
```

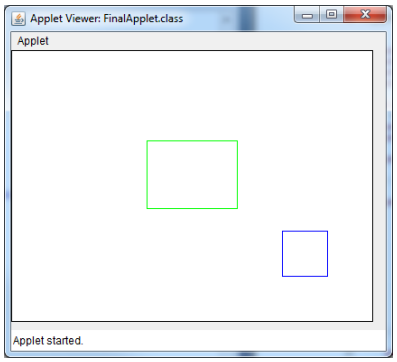
ChristmasList Class Code

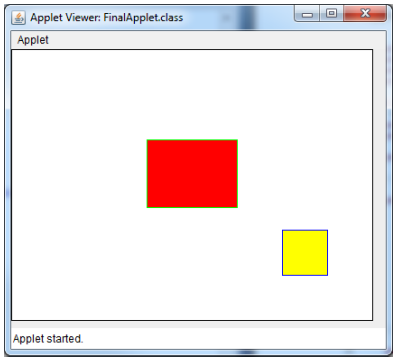
5. Below is the beginning of an applet and several screen-shots of its run are on the right. When the program starts the applet looks like the top image, as you can see from the paint method. If the user clicks inside either the green rectangle or the blue square the applet will look like the second image. That is, the green rectangle is filled in with red and still has a green boarder and the blue square is filled in with yellow and still has its blue boarder. If the user clicks anywhere outside these two rectangles the applet will look like the third image. That is, the outside of the two rectangles will be red and the two rectangles will be filled with white. Note that in this situation the rectangles still have their green and blue boarders and the screen still has its black boarder.

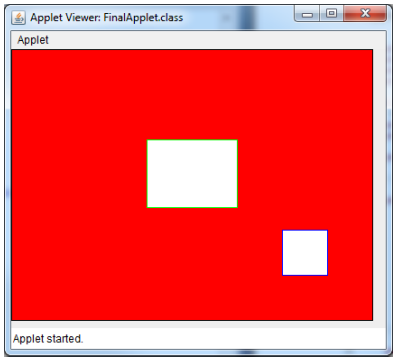
   Write the needed code for the mouseClicked event to add this functionality to the program. Please write the code on the next page.

```
1  import java.awt.*;
2  import javax.swing.JApplet;
3  import java.awt.event.*;
4  import javax.swing.*;
5
6  public class FinalApplet extends JApplet implements
       MouseListener {
7      private int drawWidth = 400;
8      private int drawHeight = 300;
9
10     public void init() {
11         addMouseListener(this);
12     }
13
14     public void paint(Graphics g) {
15         super.paint(g);
16         g.setColor(Color.WHITE);
17         g.fillRect(0, 0, drawWidth, drawHeight);
18         g.setColor(Color.BLACK);
19         g.drawRect(0, 0, drawWidth, drawHeight);
20
21         g.setColor(Color.GREEN);
22         g.drawRect(150, 100, 100, 75);
23
24         g.setColor(Color.BLUE);
25         g.drawRect(300, 200, 50, 50);
26     }
27
28     public void mouseClicked(MouseEvent evt) {
29
30     <<< Insert Code Here >>>
31
32     }
33
34     public void mousePressed(MouseEvent evt) {
35     }
36
37     public void mouseReleased(MouseEvent evt) {
38     }
39
40     public void mouseEntered(MouseEvent evt) {
41     }
42
43     public void mouseExited(MouseEvent evt) {
44     }
45 }
```

FinalApplet Run







mouseClicked Code