CS 180 Fall 2008 Final Exam

There are 30 multiple choice questions. Each one is worth 3 points. There are 4 programming questions worth a total of 110 points.

Answer the multiple choice questions on the bubble sheet given and the programming questions on the exam booklet.

Fill in the Instructor, Course, Signature, Test, and Date blanks. For "Instructor" put your **RECITATION INSTRUCTOR'S LAST NAME GIVEN BELOW**. For "Course" put CS 180. For "Test" put final.

Fill in the bubbles that correspond to your name, section and Student ID in the bubble sheet. For your section number, use the **SECTION NUMBER** for your lab and project turn in. Consult the following list:

```
0101 THU 07:30 LWSN 1106 Salman Pervez
0201 FRI 07:30 LWSN B134 Cheng Wang
0301 FRI 03:30 LWSN B134 Salman Pervez
0401 FRI 04:30 HAAS G066 Srinivas Pasupuleti
0501 FRI 09:30 LWSN B134 Ashish Gandhe
0601 FRI 02:30 LWSN 1106 Ashish Gandhe
```

For your student ID, use the 10 digit ID number on your student ID card. DO NOT USE YOUR SOCIAL SECURITY NUMBER!

Exams without names will be graded as zero. Only the answers on the bubble sheet will be counted. The questions will be discarded.

Recitation Start Time _____

Recitation TA's Name _____

Student Last Name _____

Student First Name _____

Part I. Multiple Choice Questions (3 points each):

1. Which of the following is NOT TRUE regarding error detection in your code?

   (a) Testing each small method separately will help you detect runtime errors.
   (b) Inserting print statements will help you detect syntax errors.
   (c) Stepping through your code line by line will help you detect logic errors.
   (d) None of the above.

2. What is the value of the expression below?

```
10 + 5 * 4 / 3 - 13 % 3
```

   (a) 15
   (b) 16
   (c) 15.66666666667
   (d) 12.66666666667

3. What is the output of the program below?

```
public class A
{
   public int f(int x) { return x; }
   public int f(double x) { return (int) (x * 2); }
   public double f(char x) { return x * 3; }
   public float f(int x) { return x * 4; }

   public static void main(String [] args)
   {
      System.out.print(f(1.2));
   }
}
```

   (a) 4.8
   (b) 2
   (c) There is no output due to syntax error
   (d) 3

4. Which of the following is TRUE about event-driven programming?

   (a) In a GUI program, the order in which statements are executed is determined by the events fired by objects.
   (b) An object that can fire an event can be registered with only one listener object.
   (c) If a GUI object is not registered with a listener, the program will give a run-time error.
   (d) A call to the method `actionPerformed` must be explicitly made by the programmer in order for it to handle events.

5. Given the program below, which statement should replace `//TODO#1` in order to get the output `"1,0"`?

```
public class A
{
   private static int x;
   private int y;

   public A() { x = 0; y = 0;}
   public A(int x, int y) { setX(x); setY(y); }
   public A(A a) { x = a.getX(); y = a.getY(); }
   public static void setX(int newx) { x = newx; }
   public void setY(int newy) { y = newy; }
   public static int getX() { return x; }
   public int getY() { return y; }

   public static void main(String [] args)
   {
      A a1 = new A();
      A a2 = new A(1, 2);
      a1.setX(3);
      A a3 = new A(a2);
      a3.setY(4);
      //TODO#1
      System.out.println(a2.getX() + "," + a2.getY());
   }
}
```

   (a) `a2 = new A(a3.getY() - a3.getX(), a1.getY());`
   (b) `a2 = new A(a1);`
   (c) `a2 = new A(a1.getX(), a3.getX() - a2.getX());`
   (d) `a2.setY(a1.getY());`

6. With the BorderLayout manager `pane.add(label, BorderLayout.CENTER);` is equivalent to which of the following?

   (a) `pane.add(label, CENTER);`
   (b) `pane.add(label, BorderLayout.NORTH);`
   (c) `pane.add(label);`
   (d) `pane.add(label, BorderLayout.MIDDLE);`

7. Which of the following is TRUE about action-listeners?

   (a) A sub-class of an action-listener can override the `actionPerformed` method.
   (b) The `actionPerformed` method can be defined to be static.
   (c) The `actionPerformed` method in an action-listener can return an integer value.
   (d) A class that does not implement `ActionListener` cannot define an `actionPerformed` method.

8. What is the output, when the following piece of code is executed?

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

for (int i = 0; i < a.length; ++i)
{
   System.out.print(a[i--]);
   a[++i] = -3;
}
```

   (a) 012
   (b) -3-3-3
   (c) The execution causes an ArrayIndexOutOfBoundsException
   (d) 12-3

9. Which of the following is FALSE about checked and unchecked exceptions.

   (a) They are both subclasses of `Exception`.
   (b) They may both be either caught in a catch block or declared in a throws clause.
   (c) Unchecked exceptions must be either caught in a catch block or declared in a throws clause.
   (d) Checked exceptions must be either caught in a catch block or declared in a throws clause.

10. Refer to the following code for this question. How many times is the inner loop executed?

```
for (int i = 0; i < 5; i++)
{
   for (int j = 3; j < 5; j++)
   {
      if (i == j)
         break;
   }
}
```

   (a) 10
   (b) 9
   (c) 8
   (d) 7

11. What is wrong with the following Java statement?

```
ArrayList <double> temperature = new ArrayList <double> ();
```

   (a) It should be `ArrayList temperature = new ArrayList ();`
   (b) The type parameter must be a Class.
   (c) There is nothing wrong with the `ArrayList` declaration.
   (d) `()` is not allowed, size must be declared explicitly.

12. What is the output of the following program?

```
public class Thinking
{
   private String message;

   public Thinking ()
   {
      message = "The break is close!!";
   }

   public void ChangeMessage (String str)
   {
      message = str;
   }

   public void ModifyMessage (String str)
   {
      str = "The final is not over yet...";
   }

   public static void main (String[] args)
   {
      Thinking obj = new Thinking();
      String t = "I will succeed!";

      obj.ModifyMessage(obj.message);
      System.out.print(obj.message);

      obj.ChangeMessage(t);
      System.out.println(obj.message);
   }
}
```

(a) `The final is not over yet...The final is not over yet...`
(b) `The break is close!!The break is close!!`
(c) `The break is close!!I will succeed!`
(d) `The final is not over yet...I will succeed!`

13. Which of the following statements is NOT CORRECT?

(a) The Java statement `int value = 18.0;` leads to a compile time error.
(b) The Java statement `int value = Integer.parseInt(180);` leads to a compile time error.
(c) The Java statement `int value = "180";` leads to a compile time error.
(d) The Java statement `int value = Integer.parseInt("String");` leads to a compile time error.

14. What is the output of the following program?

```
public class NullTester
{
   public void test(Object o)
   {
       System.out.print("Object Version");
   }

   public void test(String s)
   {
      System.out.print("String Version");
   }

   public static void main(String args[])
   {
       NullTester q = new NullTester();
       q.test(null);
   }
}
```

(a) There is no output due to null pointer Exception

(b) There is no output due to compile time error

(c) `String Version`

(d) `Object Version`

15. Which of the following statements is the most appropriate regrading recursion?

(a) We should not use recursion because iteration is always more efficient.

(b) We write less code if we use iteration instead of recursion.

(c) Recursion is always the most efficient technique to solve problems.

(d) In recursion, we divide a problem into subtasks where one subtask can be a smaller version of the original problem.

16. If `a` and `b` are declared as follows:

```
double[] a = {0.1,0.2,0.3};
double[] b = {0.1,0.2,0.3};
```

then what is the value of the expression `(a == b)`, and why?

(a) False, because numerical round-off error means that the floating point numbers cannot be stored exactly.

(b) True, because both arrays have the same type, size and values.

(c) A runtime error occurs because `==` can only be used for primitive types.

(d) False, because the two arrays are different objects, even though they have identical elements.

17. A string is a palindrome if it reads the same forward and backward. For the recursive method `IsPalindrome()` (defined below) to run properly, the `BOOLEAN EXPRESSION` of the base case in the given code should be ...

```
public boolean IsPalindrome (String s)
{
   if ( BOOLEAN EXPRESSION ) // base case
      return true;
   else
   {
      if (s.toLowerCase().charAt(0) == s.toLowerCase().charAt(s.length() - 1))
         return isPalindrome(s.substring(1, s.length() - 1));
      else
         return false;
   }
}
```

(a) `s.charAt(0) == s.charAt(s.length() - 1)`

(b) `s.length() <= 1`

(c) `s.charAt(0) == s.charAt(s.length())`

(d) `s.length() == 0`

18. What's the output of the following code?

```
public class Test
{
   public static void printInfixOrder(int n)
   {
      if (n < 10)
      {
         printInfixOrder(2 * n);
         System.out.print(n + " ");
         printInfixOrder(2 * n + 1);
      }
   }

   public static void main(String [] args)
   {
      printInfixOrder(1);
   }
}
```

(a) `8 4 9 2 5 1 6 3 7`

(b) `1 2 3 4 5 6 7 8 9`

(c) `8 9 4 5 2 6 7 3 1`

(d) `1 2 4 8 9 5 3 6 7`

19. Which of the following methods will correctly calculate the maximum value in an array? All of these methods compile correctly, so you should only look for logic errors. Note: Math.max(x,y) returns the maximum of its two arguments.

```
public int max1(int[] a)
{
   int maxPos = 0;
   for (int i=1; i<a.length; i++)
   {
      if (a[i] > a[maxPos]) {
         maxPos = i;
      }
   }
   return a[maxPos];
}

public int max2(int[] a)
{
   int max = 0;
   for (int i=0; i<a.length; i++) {
      if (a[i] > max) {
         max = a[i];
      }
   }
   return max;
}

public int max3(int[] a)
{
   for (int i=1; i<a.length; i++) {
      a[i] = Math.max(a[i],a[i-1]);
   }
   return a[a.length-1];
}
```

(a) max1, max2, and max3
(b) max2 and max3 only
(c) max1 and max2 only
(d) max1 and max3 only

20. Which one of the following statements is TRUE about abstract classes?

(a) You cannot create an object of an abstract class
(b) All methods of an abstract class are abstract
(c) An abstract class should be declared as final
(d) You cannot have a method parameter that is an abstract class type

21. Given the ArrayList declaration below:

```
ArrayList<Integer> a = new ArrayList<Integer>();
```

What is the output for the following operations on the variable `a`?

```
for (int i = 0; i < 3; i++)
{
   a.add(i);
}

a.add(0, 3);
System.out.println(a.get(3));
```

(a) `1`
(b) `0`
(c) `3`
(d) `2`

22. Use the following class definition for both questions 22 and 23.

```
class Node
{
   int  info;
   Node next;
}
```

Assume `Node temp` references the last element of the linked list. Which of the following conditions is TRUE about `temp`?

(a) `temp == head`
(b) `temp.next == null`
(c) `temp == null`
(d) `temp.info == 0`

23. Assume `Node temp` is currently set to the head of the linked list. Which of the following while loops is best suited to iterate through each element of the linked list?

(a)
```
while (head != null)
       head = head.next;
```
(b)
```
while (temp != null)
       head = head.next;
```
(c)
```
while (head != null)
       head = temp.next;
```
(d)
```
while (temp != null)
      temp = temp.next;
```

24. What is the output of the following piece of code?

```
int num = 5;
if (num != 5 & num++ != 6 | (num = num--) == 6)
   System.out.println("true " + num);
else
   System.out.println("false " + num);
```

(a) `false 6`
(b) `false 5`
(c) `true 6`
(d) `true 5`

25. Which of the following statements are TRUE regarding static class methods?

I. Static class methods cannot access non-static class variables.

II. Static class methods can be called by using either an object of that class type or the class name.

III. Static class methods can call non-static private class methods directly.

(a) I only
(b) II and III
(c) I and III
(d) I and II

26. Recall that in an applet, a JMenuBar contains a number of JMenus, and a JMenu contains some number of JMenuItems. Which of the following is TRUE of the way these objects are related?

(a) JMenuItem inherits from JMenu.
(b) JMenuBar inherits from JMenuItem.
(c) JMenu inherits from JMenuItem.
(d) JMenuItem inherits from JMenuBar.

27. Which of the following is TRUE of inner classes?

(a) They can be accessed from any class.
(b) They must be placed in a file with the same name as the class name.
(c) They cannot throw or catch exceptions.
(d) They have access to the private variables of the outer class.

28. How is control flow handled in event driven programs?

    (a) It is determined by the programmer.
    (b) There is no concept of control flow in event driven programs.
    (c) It is directed by the user.
    (d) It is determined by the inheritance hierarchy of window components.

29. Which of the following is INCORRECT about the GUI components in Java, strut and glue?

    (a) Strut and glue components separate visible items.
    (b) Glue is used to stick things together.
    (c) Strut can be either horizontal or vertical.
    (d) Strut and glue components are both invisible components.

30. Select the recursive function which defines the sequence 5, 8, 11, 14, 17, 20 ...

(a)
```
public int s(int n)
{
    if (n == 0)
       return 5;
    else
       return (3*n + 5);
}
```

(b)
```
public int s(int n)
{
    if (n == 0)
       return 5;
    else
       return (s(n) + 3);
}
```

(c)
```
public int s(int n)
{
    if (n == 0)
       return 5;
    else
       return s(3*n + 5);
}
```

(d)
```
public int s(int n)
{
    if (n == 0)
       return 5;
    else
       return 3 + s(n-1);
}
```

Part II. Programming Questions (110 points total):

1. (25 points) Your task is to design the cruise control system for a car. This is an event driven system which has the responsibility of keeping the car at some specified speed until switched off. This system should have no effect on the car unless explicitly switched on. You must write a class called `CruiseControl` which provides the following functions.

```
public static void Set(int newSpeed); // Set speed of car to newSpeed
public static void Resume();          // Set speed of car to current cruising speed
public static void Toggle();          // Set system to on if it's off and vice versa
```

In addition, you must clearly declare and initialize any variables required by your class.

In order to write these functions you can assume there is a `Car` class with the following functions:

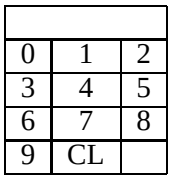
```
public static void IncreaseSpeed(int speed); // increase speed to the given speed
public static void DecreaseSpeed(int speed); // decrease speed to the given speed
public static int GetSpeed();                // return current speed of the car
```

Note: Write your code on the next page. It must be a complete class.

Solution for programming question 1:

```
public class CruiseControl
{
    private static boolean isOn = false;
    private static int speed = 0;

    public static void Set(int newSpeed)
    {
        if(!isOn)
            return;

        int carSpeed = Car.GetSpeed();

        if(newSpeed < carSpeed)
        {
            Car.DecreaseSpeed(newSpeed);
        }
        else if(newSpeed > carSpeed)
        {
            Car.IncreaseSpeed(newSpeed);
        }
        speed = newSpeed;
    }

    public static void Resume()
    {
        if(!isOn)
            return;

        int carSpeed = Car.GetSpeed();

        if(speed < carSpeed)
        {
            Car.DecreaseSpeed(speed);
        }
        else if(speed > carSpeed)
        {
            Car.IncreaseSpeed(speed);
        }
    }

    public static void Toggle()
    {
        isOn = !isOn;
    }
}
```

2. (30 points) Write a simplified `Calc` class to implement the display functionality of a calculator. You should arrange the components roughly like the image below where there is a text field for display and 11 buttons labeled 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, and CL arranged in a 4 X 3 grid. The CL button represents the Clear button in a calculator.



Your GUI must support the following:

   (a) The text field should show the appropriate number produced by a sequence of button pushes. For example, if '5', '7' and '0' are pushed in that order, the number displayed should be '570'.

   (b) The CL button should reset the number to 0.

Below is some skeleton code:

```
import java.util.*;
import javax.swing.*;
import java.awt.event.*;
import java.awt.*;

public class Calc extends JFrame implements ActionListener
{
   private int numShown;
   JTextField displayJText;

   public void display()
   {
      // TODO#1
   }

   public void actionPerformed(ActionEvent e)
   {
      // TODO#2
   }

   public static void main(String args[])
   {
      Calc calc = new Calc();
      calc.display();
   }
}
```

Note: Complete the Calc class on the next page.

Solution for programming question 2:

```
public void display()
{
    setSize(200,200);
    Container pane = this.getContentPane();

    //Adding text field
    numShown = 0;
    displayJText = new JTextField("" + numShown);
    pane.add(displayJText, BorderLayout.NORTH);

    //Adding the digit buttons
    JPanel buttonPanel = new JPanel();
    buttonPanel.setLayout(new GridLayout(4,3));
    for(int i=0;i<=9;++i)
    {
        JButton jb = new JButton("" + i);
        jb.addActionListener(this);
        buttonPanel.add(jb);
    }

    //Adding the clear button
    JButton jb = new JButton("CL");
    jb.addActionListener(this);
    buttonPanel.add(jb);

    pane.add(buttonPanel);
    setVisible(true);
}

public void actionPerformed(ActionEvent e)
{
    String command = e.getActionCommand();
    if(Character.isDigit(command.charAt(0)))
    {
        numShown = numShown*10 + Integer.parseInt(command);
        displayJText.setText("" + numShown);
    }
    if(command.equals("CL"))
    {
        numShown = 0;
        displayJText.setText("" + numShown);
    }
}
```

3. (25 points) You are given an N X N grid. You start from the southwest corner and want to reach the northeast corner. At each step, you can only go to the north neighbor or the east neighbor. How many ways do you have to reach the destiny?

   For example, suppose N is 3, you will have a grid that looks like the following where each point is labeled to help with explanation.

```
GHI
DEF
ABC
```

   You start from A and want to reach I. At the first step, you can only go from A to D (the north neighbor) or B (the east neighbor). From B, you can only go to E or C, ... So, in total, there are 6 ways to go from A to I:

```
A-B-C-F-I
A-B-E-F-I
A-B-E-H-I
A-D-E-F-I
A-D-E-H-I
A-D-G-H-I
```

   HINT: The number of ways to reach I is the sum of the number of ways to reach H and the number of ways to reach F.

   Write a class called `Grid` with a `main` method which prompts for the dimension of the grid, N (a number between 2 and 10), and output ONLY the number of possible paths. **Note: You don't need to output the paths, ONLY the number of paths.** You MUST solve this problem in the recursive way. Write a `recurSol` method to solve it using recursion. The prototype of the method is given below:

```
public static long recurSol ( int row, int column )
```

   Do NOT write any additional methods besides `main` and `recurSol`.

   Note: Write your code on the next page. It must be a complete class.

Solution for programming question 3:

```
import java.util.*;

public class Grid
{
    public static void main(String [] args)
    {
        Scanner keyboard = new Scanner(System.in);
        int n;

        System.out.println("Please input N: ");
        n = keyboard.nextInt();
        System.out.println("There are " + recurSol(n, n) + " ways.");
    }

    public statie long renSrSol ( int row, int column )
    {
        if (row == 1 || column == 1)
        {
            return 1;
        }
        else
        {
            return recurSol(row - 1, column) + recurSol(row, column - 1);
        }
    }
}
```

4. (30 points) You are given two sorted linked lists pointed by `head1` and `head2`. Create a new linked list merging the two lists such that the new linked list contains the elements from both lists arranged in sorted order. The merge function should return the head node for the newly created linked list.

   The following is the Node class used to create a linked list. `key` contains the data used for sorting, and `next` points to the next node in the linked list. You have to complete the method `merge`. 5 points will be given for commenting your code.

```
public class Node
{
   public int key;
   public Node next;

   public Node(int k, Node N)
   {
      key = k;
      next = N;
   }

   public Node merge(Node head1, Node head2)
   {
      Node N1 = head1;
      Node N2 = head2;
      Node head3 = null;

      // TODO#1

      return head3;
   }
}
```

   Note: Complete the merge method on the next page.

Solution for programming question 4:

```
public Node merge(Node head1, Node head2)
{
    Node N1 = head1;
    Node N2 = head2;
    Node head3 = null;

    Node temp1 = null;
    Node temp2 = null;

    if(N1.key<=N2.key)  // set the header of the new LL
    {
        head3 = new Node(N1.key,null);
        temp1 = head3;
        N1 = N1.next;
    }
    else
    {
        head3 = new Node(N2.key,null);
        temp1 = head3;
        N2 = N2.next;
    }

    //iterate through 1st LL and 2nd LL to add keys to new LL
    while(N1!=null || N2!=null)
    {
        if(N1==null)
        // if 1st LL is null, add all nodes from 2nd LL at the end of new LL
        {
            temp2 = new Node(N2.key, null);
            temp1.next = temp2;
            temp1 = temp2;
            N2 = N2.next;
        }
        else if(N2 == null)
        // if 2nd LL is null, add all nodes from 1st LL to the new LL
        {
            temp2 =  new Node(N1.key,null);
            temp1.next = temp2;
            temp1=temp2;
            N1 = N1.next;
        }
        else if(N1.key<=N2.key)
        // if Node in 1st LL has smaller key, add it to the new LL and advance 1st LL
        {
            temp2 =  new Node(N1.key,null);
            temp1.next = temp2;
            temp1=temp2;
            N1 = N1.next;
        }
        else
        // if Node in 2nd LL has smaller key, add it to the new LL and advance 2nd LL
        {
```

```
            temp2 =  new Node(N2.key,null);
            temp1.next = temp2;
            temp1=temp2;
            N2 = N2.next;
        }
    }
    return head3;
}
```