Computer Sciences 302
Final Exam, 20%

Saturday 12/17, 2011

Print last name: ____, first: __________________

Signature: ____ CS login: __________________

Circle Your Lecture	**Lec 1** Skrentny	**Lec 2** Skrentny	**Lec 3** Dan	**Lec 4** Peter	**Lec 5** Dalibor	**Lec 6** Rob	**Lec 7** Alicia	**Lec 8** Noah

This exam is composed of two parts. *Part I is to be answered by filling in your choice using a #2 pencil on the separate answer sheet. Part II is to be answered by writing your answers in this examination booklet.*

Before you Begin:

(1) Take a separate answer sheet and write your UW student ID number on it.
(2) Turn in your UW student ID.
(3) **On the separate answer sheet:**
- Fill in the bubbles corresponding to each digit of your UW student ID number.
- Write your name and then fill in the bubbles corresponding to each letter.
- In the "Special Codes" section under letter "A" write your lecture number and fill in the corresponding bubble, and under letter "B" write P but do not fill in a bubble.

(4) **On this examination booklet:**
- Print and sign your name above.
- Write your CS login and circle your lecture above.

(5) Check that there is a total of 12 pages followed by a spare sheet in this examination booklet.
(6) You may not use notes, books, any electronic devices, or neighbors on this exam. Turn off and put away your cell and other electronic devices now.
(7) **You have 2 hours to complete the exam**. Use your time wisely.
(8) We can't provide hints but if you need an exam question clarified or feel there is an error, please bring this to our attention. If needed, **corrections will be written on the board**.

When you've Finished:

(9) For Part I, double check that you have correctly marked the bubbles on your answer sheet. Only marks on the answer sheet matter. Marks in this examination booklet don't count.
(10) Turn in this examination booklet and your answer sheet, and make sure we return your ID.

Parts	Number of Questions	Question Format	Possible Points	Score
I	21 (1 bonus question)	Multiple Choice	63 (3 bonus points)	
II	2	Written Answers	20	
		Total	80	

Exam Reference Page

Methods from the `java.io.PrintWriter` class:

```
   PrintWriter(String filename) throws FileNotFoundException
                            // Creates a PrintWriter for the given file name
   PrintWriter(File out)    // Creates a PrintWriter from out
   void close()             // Closes the stream and any associated file
   void print(String s)     // Prints given string
   void println(String s)   // Prints given string followed by a newline
```

Methods from the `java.lang.String` class: (*REMEMBER `Strings` use 0-based indexing)

```
   String(String str)       // create a String object given another String
   int length()             // returns # of characters in this string
   char charAt(int index)   // returns the character at the zero-based index
   int compareTo(String s)  // Returns a negative if this object comes before s
                            // alphabetically, 0 if they are equal, and a positive if
                            // this object comes after s alphabetically
   boolean equals(String s)// returns true if the contents of the String
                            // instance is the same as that of the string s
```

Methods from the `java.util.ArrayList` class (`E` is the type of element):

```
   ArrayList<E>()           // Creates an empty array list to store E
   boolean add(E m)         // Appends m to the end of the list
                            // and returns true if and only if the list is changed
   void add(int index, E m)// Inserts m at the specified index
   E  get(int index)        // Returns the E at the specified index
   int indexOf(Object m)    // Returns the index of the first occurrence of m in this
                            // list, otherwise returns -1 if m is not present
   boolean isEmpty()        // Returns true if and only if this list has no elements
   int size()               // Returns the number of elements in this list
   E  remove(int index)     // Removes and returns the E at the specified index
                            // in this list
   boolean remove(Object m)// Removes the first occurrence of m from this list
                            // and returns true if and only if the list is changed
```

Methods from the `java.util.Random` class:

```
   Random()                 // Creates a new random number generator.
   int nextInt()            // Returns the next pseudorandom int value.
   int nextDouble()         // Returns the next pseudorandom double value
                            // between 0 (inclusive) 1.0 (exclusive).
```

Methods from the `java.util.Scanner` class:

```
   Scanner(System.in)       // Creates a Scanner object that reads from the keyboard
   Scanner(File in)         // Creates a Scanner object that reads from in
   void close()             // Closes the scanner and any associated file
   *Note: Tokens are sequences of characters separated by whitespace.
   boolean hasNext()        // Returns true iff another token exists in input
   String next()            // Returns the next token of input
   boolean hasNextInt()     // Returns true iff another int exists in input
   boolean hasNextDouble()  // Returns true iff another int exists in input
   boolean hasNextLine()    // Returns true iff another line exists in input
   String nextLine()        // Returns the next line of input
   *Note: Methods below throw an InputMismatchException if the input type is wrong.
   int    nextInt()         // Scans the next token of the input as an int
   String next()            // returns the next token of input
   double nextDouble()      // Scans the next token of the input as a double
```

The `Comparable` interface:

```
   int compareTo(Object ob)// Returns a negative if this object is less than ob,
                            // 0 if they are equal, and a positive if this object
                            // is greater than ob
```

Part I Multiple Choice *[21 questions, 3 points each, 63 total points]*

There are 21 question each worth 3 points with a maximum score of 63 points **(there is 1 bonus question)**.
For the questions on the following pages, **choose the one best answer after reading all of the choices**.
Use a #2 pencil to fill in the bubble on your answer sheet that corresponds to your answer for each question.

1.) Consider the following program:

```
public class Methods {

   public static void main(String[] args) {
      System.out.print("main,");
      m3();
      m2();
      System.out.println("bye!");
   }

   public static void m1() {
      System.out.print("m1,");
   }
   public static void m2() {
      m1();
      System.out.print("m2,");
   }
   public static void m3() {
      m1();
      m2();
      System.out.print("m3,");
   }
}
```

Which one of the following shows what the `Methods` program displays when executed?

A. `main,m1,m1,m1,m2,m2,m3,bye!`
B. `main,m1,m1,m2,m3,m1,m2,bye!`
C. `main,m1,m2,m1,m1,m2,bye!`
D. `main,m1,m2,m1,m1,m2,m3,bye!`
E. `main,m2,m1,m3,m1,m2,m1,bye!`

2.) Consider the following method that is passed an array of integers and two additional integers that are guaranteed to be within the bounds of the array:

```
public static boolean method(int[][] a, int r, int c) {
   for (int i = 0; i < a[r].length; i++) {
      if (a[r][c] != a[r][i]) { return false; }
   }
   for (int i = 0; i < a.length; i++) {
      if (a[r][c] != a[i][c]) { return false; }
   }
   return true;
}
```

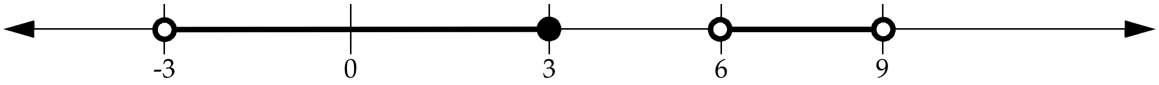
Which one of the following *best* describes the circumstances under which this method returns true?

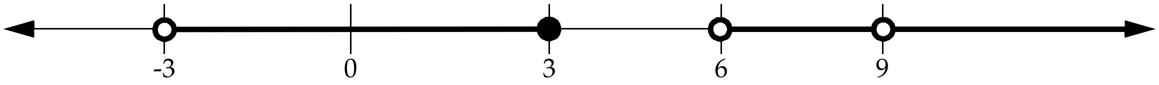
A. always returns true
B. if and only if all of the values in row `r` are same as the value in the element at `r,c`
C. if and only if all of the values in column `c` are same as the value in the element at `r,c`
D. if and only if the values in row `r` and in column `c` are same as the value in the element at `r,c`
E. if and only if the values in row `r` and in column `c` are *not* same as the value in the element at `r,c`

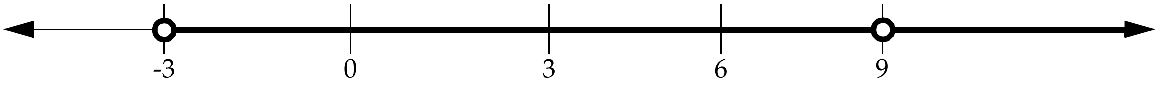
3.) Consider the following condition where `n` is an integer variable:

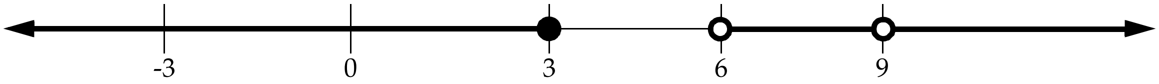
```
((n > -3) && (n <= 3)) || ((n > 6) && (n != 9))
```

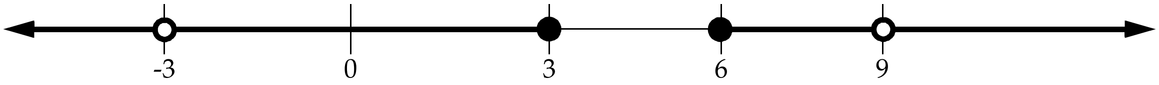
Which one of the following number lines correctly shows in bold the range of `n` where the condition evaluates to `true`? Note filled circles represent points that are included in the range and hollow circles represent points that are not included.

A.


B.


C.


D.


E.


4.) Which one of the following statements about object-oriented programming is *false*?

A. An instance variable is made `private` to prevent direct access by code in other classes.
B. A class variable must be `static` and it's shared among all instances of the class.
C. Constructors are used to initialize an object's instance variables and instance constants.
D. Instance methods must be `public` if they are to be used by code in other classes.
E. Class (`static`) methods get a `this` reference during execution but cannot access instance variables.

5.) Consider the following incomplete class:

```
public class Tag {

   private MODIFIER int n = 0;
   private final int number;

   public Tag(PARAMETER) {
      STATEMENT(S)
      //additional constructor code not shown...
   }
```

Which one of the following replacements for `MODIFIER`, `PARAMETER` and `STATEMENT(S)` when used to complete the code above guarantees that each instance of `Tag` will have a unique value for `number`?

	MODIFIER	PARAMETER	STATEMENT(S)
A.	`/*nothing*/`	`/*nothing*/`	`n++; number = n;`
B.	`final`	`int uniqueNumber`	`number = uniqueNumber;`
C.	`final`	`/*nothing*/`	`n++; number = n;`
D.	`static`	`int uniqueNumber`	`number = uniqueNumber;`
E.	`static`	`/*nothing*/`	`n++; number = n;`

6.) Consider the following code fragment where `density` is an integer variable:

```
if (density > 5) {
   System.out.print("Normal");
}
else if (density >= 8) {
   System.out.print("Above Normal");
}
else {
   System.out.print("Abnormal");
}
```

Which one of the following statements about this code fragment is true?

A. "`Above Normal`" will never be displayed.
B. "`Above Normal`" is displayed if and only if `density` is greater than or equal to 8.
C. "`Abnormal`" is displayed if and only if `density` is less than 5.
D. "`Abnormal`" is displayed if and only if `density` is less than 8.
E. "`Normal`" is displayed if and only if `density` is greater than or equal to 5.

7.) Consider the following method:

```
public static void setArray(char[][] m) {
   for (int c = 0; c < m.length; c += 2) {
      for (int r = 0; r <= c; r++) {
         m[r][c] = '?';
      }
   }
}
```

Assume that `m` is a 4-by-4 array of characters initialized to contain only spaces. Which one of the following correctly illustrates the values in `m` after it has been set by this method?

A.

?			
?	?	?	

B.

?		?	
		?	
		?	

C.

?	?		
?	?	?	?

D.

	?		?
	?		?
			?
			?

E.

?			
?	?		
?	?	?	
?	?	?	?

8.) If multiplication had lower precedence than addition the result of the expression:

```
2 * 3 + 4 * 3 + 2
```

would be which one of the following?

A. 14
B. 20
C. 32
D. 70
E. 144

The questions on this page are based on the following code and each question has only FOUR choices:

```
class Car {
   private int     speed;
   private double position;
   public  Car(int s, double p)  { speed = s; position = p;                   }
   public  int     getSpeed()     { return speed;                              }
   public  double getPosition()  { return position;                           }
   public  boolean equals(Car c) { /*RETURNS true IFF c IS EQUAL TO this CAR*/ }
}
class Lane {
   private ArrayList<Car> cars;
   public  Lane()                 { cars = new ArrayList<Car>();               }
   public  Car     getCar(int i)  { return cars.get(i);                        }
   public  int     getSize()      { return cars.size();                        }
   //assume other code not shown, e.g., to add cars
}
class Highway {
   private Lane[] lanes;
   public  Highway(int n)         { /*CONSTRUCTS A HIGHWAY WITH n LANES */      }
   public  Lane    getLane(int ln){ return lanes[ln];                          }
   public  int     size()         { return lanes.length;                       }
   //assume other code not shown, e.g., to add cars
}
```

9.) How many `Lane` objects are constructed if the body of the `Highway` constructor is:

```
lanes = new Lane[n]; //ASSUME n IS A POSITIVE INTEGER
```

A. zero
B. one
C. `n`
D. `n - 1`

10.) If `h` is a properly constructed `Highway` object in the main method of a `Game` class and each lane within `h` contains cars, which one of the statements below displays the speed of the first car in the last lane?

A. `System.out.println(h.getLane(h.size).getCar(1).getSpeed());`
B. `System.out.println(h.getLane(h.size()-1).getCar(0).getSpeed());`
C. `System.out.println(h.lanes[h.size()-1].getCar(0).getSpeed);`
D. `System.out.println(h.lanes[lanes.length-1].getCar(1).getSpeed());`

11.) Assume the method below is added to the `Highway` class and `givenCar` is a properly constructed `Car`.

```
public int contains(Car givenCar) {
   for (int ln = 0; ln < lanes.length; ln++) {
      for (int c = 0; c < EXPRESSION; c++) {
         if (CONDITION) { return ln; }
      }
   }
   return -1;
}
```

Which one of the following replacements for `EXPRESSION` and `CONDITION` when used to complete the method above will return the lane number of the given car or -1 if it is not found?

	EXPRESSION	CONDITION
A.	`lanes.length`	`lanes[ln].getCar(c).equals(givenCar)`
B.	`lanes.length`	`lanes[ln].getCar(givenCar).equals(c)`
C.	`lanes[ln].getSize()`	`lanes[ln].getCar(c).equals(givenCar)`
D.	`lanes[ln].getSize()`	`lanes[ln].getCar(givenCar).equals(c)`

12.) Consider the following incomplete method that is passed an array of `String` objects that you may assume has been filled with names and has a length of at least one:

```
public static String findFirstName(String[] names) {
   int first = 0;
   for (int i = 1; i < names.length; i++) {
      if (names[first].compareTo(names[i])  OPERATOR  VALUE) {
         first = i;
      }
   }
   STATEMENT
}
```

Which one of the following replacements for `OPERATOR`, `VALUE` and `STATEMENT`, when used to complete the method above, returns the first name in the array based on alphabetical ordering?

	OPERATOR	VALUE	STATEMENT
A.	`<`	`0`	`return names[first];`
B.	`==`	`-1`	`return first;`
C.	`==`	`-1`	`return names[i];`
D.	`>`	`0`	`return names[first];`
E.	`>`	`0`	`return names[i];`

13.) Assume `n` is an integer variable and `done` is a boolean variable. Which one of the following pairs of expressions is ***not*** equivalent to each other (that is, doesn't evaluate to the same result)?

A. `!done` is equivalent to `done == false`
B. `!(n <= 11)` is equivalent to `n > 11`
C. `n == 11 && n == 22` is equivalent to `false`
D. `n < 22 || n > 22` is equivalent to `n != 22`
E. `!(n < 11 && n > 22)` is equivalent to `n < 11 || n > 22`

14.) Consider the following partially implemented method where the code in the `try` block might throw a `FileNotFoundException` (a checked exception):

```
public boolean processFile(String filename) {
   try {
      //code to process file not shown...
   } catch (InputMismatchException e) {
      return false;
   }
   return true;
}
```

Which of the following additions to the code above ensures that this won't result in a compiler error?

i. Add to the method header: `throws FileNotFoundException`
ii. Add to the `try` block: `throw new FileNotFoundException();`
iii. Add after the existing catch block: `catch (FileNotFoundException e) { /*...*/ }`

A. *i* only
B. *i* or *ii* only
C. *i* or *iii* only
D. *ii* or *iii* only
E. *i, ii* or *iii*

The questions on this page are based on the following *partially shown* code:

```
class Exceptions {

   public static void main(String[] args){
      try {
         methodX();
         //LINE A - see questions below
      } catch (NullPointerException x) {
         System.out.print("Error 1,");
      } catch (IndexOutOfBoundsException x) {
         System.out.print("Error 2,");
      }
      System.out.print("main done,");
   }

   public static void methodX() {
      methodY();
      try {
         //LINE B - see questions below
      } catch (NullPointerException x) {
         System.out.print("Error 3,");
      }
      System.out.print("methodX done,");
   }

   public static void methodY() {
      try {
         //LINE C - see questions below
      } catch (ArithmeticException x) {
         System.out.print("Error 4,");
      }
      System.out.print("methodY done,");
   }
}
```

15.) What is output if LINE A is replaced with code that will throw a `NullPointerException`:

A. `Error 1,`
B. `Error 1,main done,`
C. `methodY done,Error 1,main done,`
D. `methodY done,methodX done,Error 1,main done,`
E. The program crashes and message about the exception is displayed.

16.) What is output if LINE B is replaced with code that will throw an `IndexOutOfBoundsException`:

A. `Error 2,`
B. `Error 2,main done,`
C. `methodY done,Error 2,main done,`
D. `methodY done,methodX done,Error 2,main done,`
E. The program crashes and message about the exception is displayed.

17.) What is output if LINE C is replaced with code that will throw a `NullPointerException`:

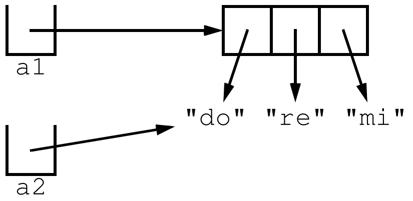
A. `Error 1,`
B. `Error 1,main done,`
C. `methodY done,Error 1,main done,`
D. `methodY done,methodX done,Error 1,main done,`
E. The program crashes and message about the exception is displayed.

18.) Consider the following code:

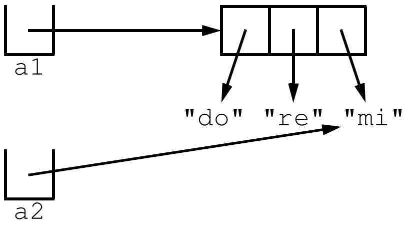
```
String[] a1 = {"do", "re", "mi"};
String[] a2 = a1;
for (int i = 0; i < a1.length; i++) {
   a2[i] = a1[i];
}
```

Which one of the following is the correct memory diagram after the lines of code above execute?

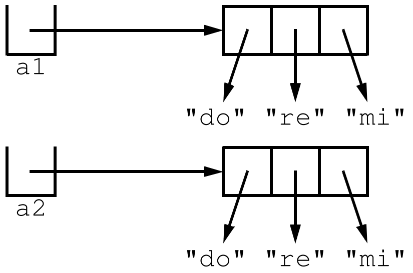
A.



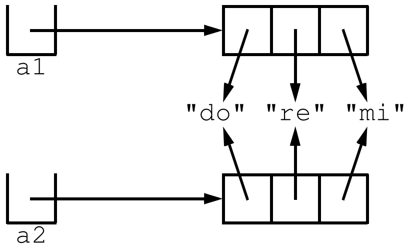
D.



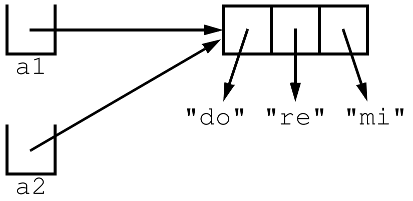
B.



E.



C.



19.) What would be displayed if the following code fragment was executed?

```
for (int i = 1; i < 7; i+=2) {
   switch (i) {
      case 1: System.out.print("C");
      case 2: System.out.print("H"); break;
      case 3:
      case 4: System.out.print("A"); break;
      case 6: System.out.print("N"); break;
      case 9:
      default: System.out.print("T");
   }
}
```

A. CAN

B. CAT

C. CHAN

D. CHAT

E. CHANT

20.) Assume the following method is passed an array of integers, named d, that is limited to have integers in the range of 0 to 9 inclusive:

```
public static boolean doSomething (int[] d) {

   boolean[] b = new boolean[10]; //elements are initialized to false

   for (int i = 0; i < d.length; i++) {
      b[d[i]] = true;
   }
   for (int i = 0; i < b.length; i++) {
      if (!b[i]) {
         return false;
      }
   }
   return true;
}
```

Which one of the following *best* describes what method doSomething does?

A. It returns true if and only if the array of integers has duplicates.
B. It returns true if and only if the array of integers has *no* duplicates.
C. It returns true if and only if the array of integers has all of the digits from 0 to 9.
D. It returns true if and only if the array of integers does *not* have all of the digits from 0 to 9.
E. It always throws an ArrayIndexOutOfBoundsException.

21.) Consider the following complete implementation of the Data class:

```
public class Data {
   private int data;
   public  int get () {
      return data;
   }
   public  void set (int d) {
      data = d;
   }
}
```

Also consider the following application class:

```
public class DataApp {
   public static void main(String[] args) {
      int data = 11;
      Data d1 = new Data();
      Data d2 = new Data();
      d1.set(22);
      data = d1.get();
      d2.set(data);
      d1.set(33);
      d1.set(44);
      System.out.println(data + ", " + d1.get() + ", " + d2.get());
   }
}
```

What is printed when the DataApp program is executed?

A. 11, 22, 33
B. 11, 44, 22
C. 22, 33, 22
D. 22, 44, 11
E. 22, 44, 22

Part II Written Answers [2 question, 20 total points]

Write your answers to the written question in this examination booklet. If you need more room use the last page and indicate by the question where your answer continues.

1.) **[10 points] Complete the `saveGrid` method below** that saves a two-dimensional (2D) array in an output file and returns true if it was successful, false otherwise. The method is passed a 2D grid and a `File` object representing the output file. The grid is saved so that the first line has the number of rows in the grid followed by the number of columns separated by a space. Each subsequent line in the file is one row of the grid with the values in that row separated by spaces. *For full credit, your method should throw an illegal argument exception if either of its parameters are invalid.*

```
public static boolean saveGrid(int[][] grid, File outfile) {
```

2.) **[10 points]** Consider a variation of the program three driving game having a highway of lanes with different classes of objects on the road (e.g., Cars, PotHoles, etc.) that can be hit (e.g., run into). Also, consider the `Hittable` interface below having the following public class constant and public method:

```
static final char CRASHED = //SOME VALUE NOT SHOWN;
boolean hasHit(Object o);  //Returns true if the front of this object
                           //hits the back of Object o,
                           //otherwise returns false.
```

Modify the `Car` class below so that it implements the `Hittable` interface. Note, if this car hits the back of another car, both cars have their state variable changed to crashed. If this car hits a pot hole, there is only a 0.15 probability that the car's state is changed to crashed.

```
public class PotHole {

   private int posOfCenter; //position of the center of the PotHole
   private int length;      //length of PotHole

   public int getCenter() {
      return this.position;
   }
   public int getLength() {
      return this.length;
   }
   //assume additional code for class is not shown and not needed
}

public class Car {

   private int  posOfFront;//position of Car's front
   private int  length;    //length of Car
   private char state;     //state of Car (e.g., crashed, stalled, moving)

   //assume additional code for class is not shown and not needed
```

Print last name: ____, first: __________________

Signature: ____ CS login: __________________

This sheet must be returned with the rest of your exam.
If you remove this sheet, fill in the information above and tuck it between the pages of your exam when you turn it in.