

# Computer Sciences 302

## Final Exam, 20%

Friday 12/21, 2012

Print last name: \_\_\_\_\_, first: \_\_\_\_\_

Signature: \_\_\_\_\_ CS login: \_\_\_\_\_

<b>Circle Your Lecture</b>	<b>Lec 1</b>	<b>Lec 2</b>	<b>Lec 3</b>	<b>Lec 4</b>	<b>Lec 5</b>	<b>Lec 6</b>	<b>Lec 8</b>
	Deb Deppeler	Deb Deppeler	Stephen Brown	Melissa Tress	Seth Pollen	Ben Miller	Shreedhar Hardikar

**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 "J" write P and fill in a bubble 1.
- (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	16	Multiple Choice	48	
II	2	Written Answers	24	
Total			72	

## 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** [16 questions, 3 points each, 48 total points]

There are 16 question each worth 3 points with a maximum score of 48 points (**there is no 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.) What is output by this code fragment?

```
char [] choice = { 'A', 'B', 'C', 'D' };
String s = "-";
for ( int i = 0 ; i < choice.length; i++ ) {
    switch ( choice[i] ) {
        case 'A': s = "A" ;
        case 'C': s = s + "C" ;
        case 'D': s = s + "D" ; break;
        default: s = "Z";
    }
}
System.out.println( s );
```

- A. ZCDD
- B. ACDZCD
- C. ACDZCDD
- D. -ACDZCDD
- E. -AZCD

2.) What is the value in n after this code fragment executes?

```
int n = 0;
for ( int i = 1 ; i < 3 ; i = i + 1 )
    for ( int j = 3 ; j < 6 ; j = j + 3 )
        n += i + j;
System.out.println( n );
```

- A. 4
- B. 7
- C. 9
- D. 11
- E. 13

3.) Which statement is *true*?

- A. An instantiable class type **may not** contain static fields or methods.
- B. A Java class with a *main* method **may** contain non static fields and methods.
- C. Object-oriented programs **may not** contain methods that return primitive type data.
- D. In Java, one **can not** create instances of a class that uses primitive types for any of its' fields.
- E. All Java classes **must have** a public static void main(String [] args) method.

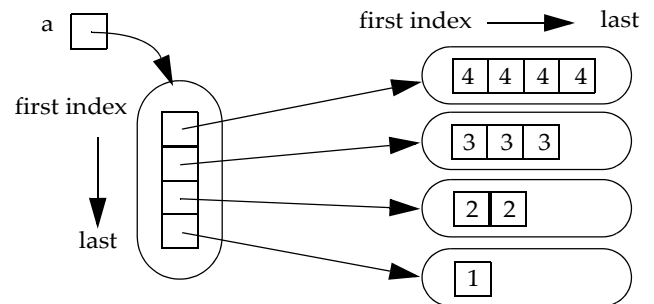
4.) What is output by this code fragment?

```
int [][] a = { {2,1}, {4,2}, {2,5}, {-2,3} } ;
String s = "";
for ( int i=0; i < a.length; i++ ) {
    int b = a[i][0];
    int c = a[i][1];
    if ( c - b >= 0 )
        s += b + " " + c + " ";
}
System.out.println( s );
```

- A. 2 1 4 2
- B. 2 1 4 2 2 5 2 -3
- C. 2 4 2 2 1 2 5 -3
- D. 1 2 2 4 2 5 -3 2
- E. 2 5 -2 3

5.) Recall that two dimensional arrays in Java are not required to have the same length for the second dimension. This diagram represents the contents of such a 2D array.

Which choice completes the code fragment represented by this object diagram?



```
int [][] a = new int[4][];
for ( int i = 0; i < 4 ; i++ )
{
    a[i] = new int[4-i];
    for ( int j = 4-i; j > 0 ; j-- ) {
        STMT ;
    }
}
```

- A. STMT
- B. a[i][j-1] = i
- C. a[i][j-1] = 4-i
- D. a[i][j-1] = 4-j

- 6.) Consider the following incomplete class for bullets, in a game that allows different bullets to have different speeds:

```
public class Bullet {
    private Position pos;
    public MODIFIER(S) double speed;
    public Bullet( Position p, double s ) {
        pos = p;
        speed = s;
        //additional constructor code not shown...
    }
}
```

Which one of the following replacements for MODIFIER(S) when used to complete the code above guarantees that any instance of `Bullet` will have the same speed, throughout its "life" in the program, that was specified at the time the bullet was created?

- A. MODIFIER(S)  
 /\*nothing\*/  
 B. final  
 C. static  
 D. static final

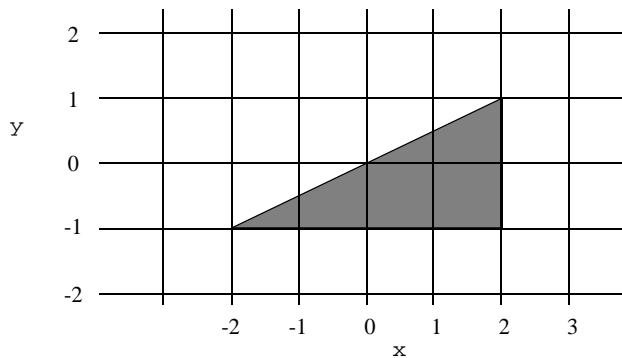
- 7.) Consider the following code fragment where `d` is an integer variable and `r` is a boolean variable:

```
String s = "T";
if ( d > 5 && r ) {
    s = "B";
}
else if ( d > 1 && !r ) {
    s = "K";
}
else {
    s = "W";
}
System.out.println( s );
```

Which one of the following statements is **true**?

- A. Only "T" will never be displayed.  
 B. Only "B" will never be displayed.  
 C. Only "K" will never be displayed.  
 D. Both "T" and "K" will never be displayed.

- 8.) Which expression correctly completes a conditional expression so that it returns true if and only if the point (x,y) would fall within the shaded area of the grid? The border of the shaded area is considered in the bounds of the area.



- A.  $y \geq -1 \ \&\& \ x \leq 2 \ \&\& \ y \leq 2*x$   
 B.  $y \geq -1 \ \&\& \ x \leq 2 \ \&\& \ y \geq 2*x$   
 C.  $y \geq -1 \ \&\& \ x \leq 2 \ \&\& \ y \leq x/2$   
 D.  $y \geq -1 \ \&\& \ x \leq 2 \ \&\& \ y \geq x/2$   
 E.  $y \geq -1 \ \&\& \ x \leq 2 \ \&\& \ y \leq x$
- 9.) Consider the following incomplete method that is passed an integer (number of cents) and returns a String showing the value in dollars and cents.

For example: `cents2dollarStr(156)` would return `$1.56` and  
`cents2dollarStr(157500)` would return `$1575.00`

```
public static String cents2dollarStr( int value ) {
    int d = DOLLARS ;
    int c = CENTS ;
    String s = "$" + d + "." + c ;
    if ( c == 0 ) s += "0";
    return s;
}
```

Which one of the following replacements for DOLLARS and CENTS, when used to complete the method above, returns the integer as a monetary value string?

- A.  $\frac{\text{DOLLARS}}{\text{value}-100}$        $\frac{\text{CENTS}}{\text{value} / 100}$   
 B.  $\text{value}/100$        $\text{value} \% 100$   
 C.  $\text{value}\%100$        $\text{value} - 100$   
 D.  $\text{value}-100$        $\text{value} \% 100$   
 E.  $\text{value}/100$        $\text{value} - 100$

10.) Consider the following program:

```
import java.io.*;
import java.util.*;
public class Q10 {
    public static void main(String[] args)
    {
        String fn = "data.txt";
        readFile(fn);
        System.out.println( "main");
    }

    public static void readFile(String fn) throws IOException {
        Scanner in = new Scanner(new File( fn ));
        int n = in.nextInt();
        for ( int i = 0 ; i < n ; i++ ) {
            if (in.hasNextDouble()) {
                in.nextDouble();
                System.out.print("m1,");
            }
            else {
                System.out.print("m2,");
            }
            in.nextLine();
        }
        in.close();
    }
}
```

What is the output that is produced, if data.txt exists and the contents of data.txt are as shown to the right?

- A. main,m1,m2
- B. main,m1,m1,
- C. m1,m2,main
- D. m1,m2,m2,main
- E. m1,m1,m2

data.txt

2 6 bananas apples
--------------------------

11.) Which declaration is **not** valid in Java? Assume any necessary import statements have been done.

- A. IllegalArgumentException e ;
- B. Integer [] integerArray ;
- C. ArrayList<int> intArrayList ;
- D. String [][] twoDStringArray ;

The questions on this page are based on the following public interface to these three classes. Assume private data and methods, and other methods exists to make them fully functional.

// ASSUME ALL METHODS WORKS AS ADVERTSTIED

Each question has only FOUR choices:

```
public class Occupant {
    public Occupant(String login, String name) { /* constructor */ }
    // other methods to access and update Occupant information
}

public class Office {
    public Office(int roomNo, String phone){ /* constructor */ }
    public void add( Occupant p ) { /* adds an Occupant to list */ }
    public String getPhone() { /* returns phone number */ }
    public Occupant[] getOccupants() { /* returns array of Occupants */ }
}

public class Building {
    // Assume a constructor properly initializes the building
    // with an array of offices.

    // returns the Office with the specified room number
    public Office getOffice(int roomNo){ /*assume it works*/ }

    // returns the office that has Occupant with the specified login name*/;
    public Office getOffice(String login){ /* assume it works */ }

    // adds a [newly created Pokemon to the database
    public void add(Office ofc) { /* assume it works; }
}
```

Assume `cs` is a properly constructed Building object for each of the following problems.

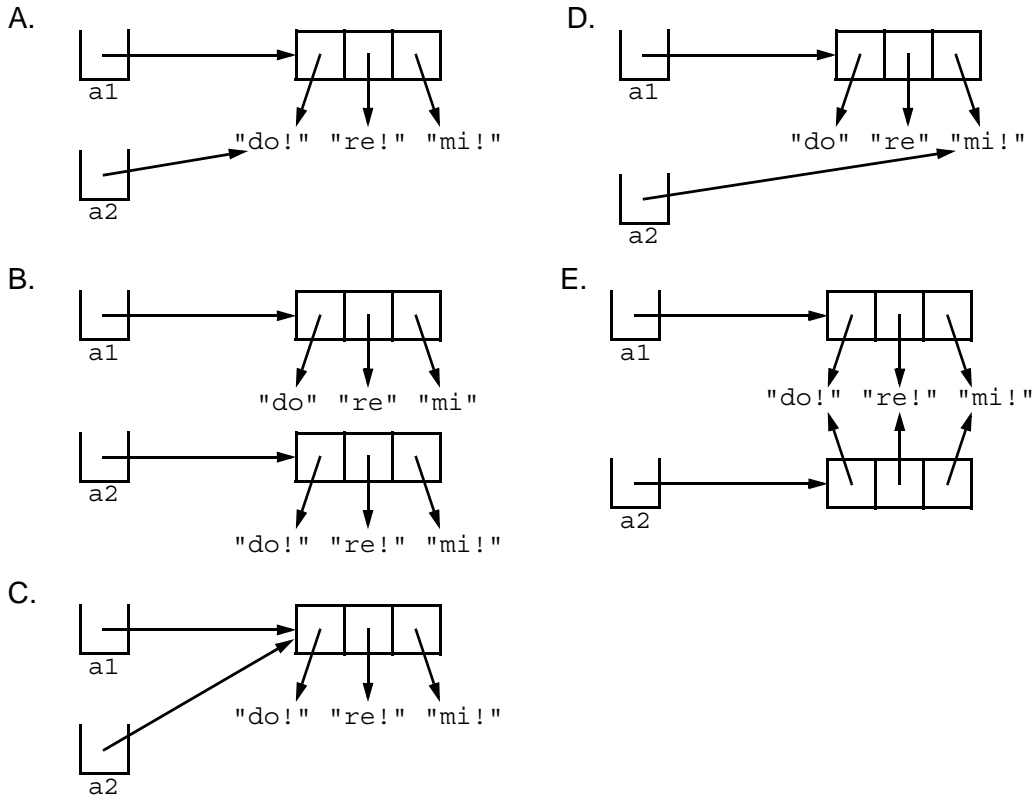
- 12.) Which one of the statements below will add Deb Deppeler to an existing Office with room number 5376?
- A. `cs.add(5376, "deppeler", "Deb Deppeler");`
  - B. `cs.add( new Office( 5376, new Occupant("deppeler", "Deb Deppeler") ) );`
  - C. `cs.add(5376, new Occupant("deppeler", "Deb Deppeler") );`
  - D. `cs.getOffice(5376).add( new Occupant("deppeler", "Deb Deppeler") );`
- 13.) Which expression will find the phone number for Seth Pollen "pollen", if Seth has previously been added as an Occupant of one of the offices in the building?
- A. `cs.getOffice("Seth Pollen").getPhone();`
  - B. `cs.getOffice("pollen").getPhone();`
  - C. `cs.getPhone(getOffice("Seth Pollen"));`
  - D. `cs.getPhone(getOffice("pollen"));`
- 14.) Which expression will return the number of occupants in office 5386?
- A. `getOccupants( cs.getOffice(5386) ).length`
  - B. `getOccupants[ cs.getOffice(5386) ].length`
  - C. `cs.getOffice(5386).getOccupants().length`
  - D. `cs.getOffice(5386).getOccupants[].length`



15.) Consider the following code:

```
String[] a1 = {"do", "re", "mi"};
String[] a2 = new String[3];
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?



16.) Assume that **n** an integer and that **d** is an array of integers:

```
for ( int i=0; i < d.length; i++ ) {
    if ( d[i]==n ) {
        int j=i;
        while ( j < d.length && d[j] == n ) {
            j++;
        }
        if ( j < d.length ) {
            d[i]=d[j];
            d[j]=n;
        }
    }
}
```

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

- A. It returns an array filled with the value n.
- B. It returns an array with all values of n removed.
- C. It returns an array with all values of n moved to start of array.
- D. It returns an array with all values of n moved to end of array.

**Part II Written Answers** [2 question, 24 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.) [12 points total] Partially define a class `Pokemon` as described below. Note: This question does not require a complete class definition, but **you must complete the parts as described below**. Assume that each exception class noted extends `RuntimeException` and has already been defined for you with a valid default no-arg constructor.

**For full credit, your class must:**

- Declare the class header correctly and be instantiable.
- Implement the `Comparable` interface so that arrays of `Pokemon` may be **sorted by height**. (sorting by either ascending or descending height is acceptable)
- Define a single public **constructor** that:
  - Accepts exactly four arguments in this order:  
a string of characters for the name, a string for the type, an integer for the height, and a string of characters for the description.
  - Will throw a `NameMustBeAllCapsException` if the name argument is not all caps. Digits and other characters are okay, but all letters must be in CAPS.
  - Will throw an `InvalidPokemonTypeException` if the specified type is not one of the following case-sensitive types: "DRAGON", "PSYCHIC", or "BUG".
  - Will throw a `NegativeHeightException` if the height is less than zero.
  - Will save only the first **256** characters of the description string.
- Declare constant(s) and private instance field(s) as needed to implement the functionality described here.

**Reminder:** Assume that the exception classes have each been previously defined to inherit from `RuntimeException` and that they each contain a valid default no-arg constructor.

*Problem 1 work continued...*

2.) [12 points] Complete the `saveHighScores` method as its described.

```
/**
 * Write the three highest scores to a file and return them in an array.
 *
 * Given an array of previous high scores and an array of current player
 * scores, compare each of the current player scores to the high score
 * and correctly update the high score array.
 *
 * Once all highest scores have been determined, write those scores to
 * output file that is connected (for you) to a PrintWriter object.
 *
 * Finally, return the array of high score values.
 *
 * Assume that the hs parameter is sorted highest to lowest and
 * that it does not and should not contain duplicates.
 *
 * @param hs An array sorted from highest to lowest without duplicates.
 * @param cs An array of current player scores that is not sorted and
 * and may contain duplicates (imagine ties were possible).
 * @param pw A PrintWriter that has been successfully connected to a
 * file for writing output.
 */
public int[] saveHighScores( int[] hs, int[] cs, PrintWriter pw ) {
```

```
} // end of saveHighScores method
```

\_\_\_\_ / 12

*Problem 2 work continued...*

**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.