

Computer Sciences 302

Midterm Exam 1, 20%

Thursday 10/20, 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
--------------------------------	--------------------------	--------------------------	---------------------	-----------------------	-------------------------	---------------------	------------------------	----------------------

Before you Begin:

- (1) Take a separate answer sheet (i.e., scantron) 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 S 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 in this exam.
- (6) You may not use notes, books, calculators (or any other electronic devices), or neighbors on this exam. Turn off and put away your cell phone, pager, pda, etc. now.
- (7) The exam is intended to take 90 minutes, but **we will give you 2 hours to complete the exam**.
- (8) We can't provide hints but if you need an exam question clarified or feel that there is an error, please bring this to our attention. If needed, **corrections will be written on the board**.

When you've Finished:

- (9) Double check that you have correctly marked the bubbles on your answer sheet. Only answers marked on your 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.


Taking the Exam

There are 25 question each worth 3 points with a maximum score of 72 points (there is a 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. Note a reference is provided on the next page, which you should review when the exam begins.

Exam Reference Page

Operator Precedence Table:

level	operator	description	
higher	(<expression>)	grouping with parentheses	
	+ - (<type>) ++ -- !	unary plus/minus type casting dec/increment logical not	
	* / %	multiplicative	
	+ -	additive	
	< <= > >=	relational	
	== !=	equality	
	&&	logical and	
		logical or	
	= += -= *= /= %=	assignment and compound assignments	
	lower		

Constant and Methods from the `java.lang.Math` class:

```

PI //represents the constant  $\pi$ 
double abs (double n) //Returns the absolute value of n.
double pow (double n, double p) //Returns  $n^p$ .
double sqrt(double n) //Returns the square root of n.

```

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

```

int length() //Returns # of characters in this String.
char charAt(int index) //Returns the character at the index.
boolean equals(String s) //Returns true if the contents of this String
//is the same as the contents of String s.
boolean equalsIgnoreCase(String s) // Returns true iff the contents of the
// this string is the same as that of the
// string s, ignoring differences in case.
String substring(int beginIndex)
//Returns a new string that is a substring of this string
//starting at beginIndex to the end of the this string.
String substring(int beginIdx, int endIdx)
//Returns a new string that is a substring of this string
//starting at beginIdx up to but not including endIdx.

```

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

```

Random() //Creates a new random number generator.
Random(int s) //Creates a new random number generator seeded with s.
int nextInt() //Returns the next pseudo-random integer value.
int nextInt(int n) //Returns the next pseudo-random integer value
//between 0 (inclusive) and n (exclusive).

```

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

```

Scanner(System.in) //Creates a Scanner object that reads from the keyboard.
boolean hasNextInt() //Returns true if the next input would be an integer.
int nextInt() //Returns the next input as an integer.
String nextLine() //Returns the next input line as a String.

```

- 1.) Consider the following incomplete method that is passed an index named `pos` and a *partially-filled* array of integers named `list` with `size` guaranteed to be number of used elements in the array:

```
public static boolean removeValue(int pos, int[] list, int size){
    if (CONDITION) {
        return false;
    }
    for (int i = pos; i < LIMIT; i++){
        array[i] = array[i+1];
    }
    STATEMENT
}
```

Which one of the following replacements for CONDITION, LIMIT and STATEMENT, when used to complete the code above, results in the value at index `pos` being removed from `list`?

	<u>CONDITION</u>	<u>LIMIT</u>	<u>STATEMENT</u>
A.	<code>pos < 0 pos >= size</code>	<code>size</code>	<code>size--;</code>
B.	<code>pos < 0 pos >= size</code>	<code>size - 1</code>	<code>return true;</code>
C.	<code>pos < 0 pos >= size</code>	<code>size - 1</code>	<code>size--;</code>
D.	<code>pos < 0 && pos >= size</code>	<code>size</code>	<code>size--;</code>
E.	<code>pos < 0 && pos >= size</code>	<code>size - 1</code>	<code>return true;</code>

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

```
int n = 5;
if (n > 0) {
    System.out.print(n);
    n--;
}
```

- A. 5
 B. 43210
 C. 54321
 D. 4
 3
 2
 1
 0
 E. 5
 4
 3
 2
 1

- 3.) Consider the following code:

```
for (int i = 1; i <= 20; i+=2) {
    System.out.println("Repeat");
}
```

How many times is the word "Repeat" displayed by this code fragment?

- A. 9
 B. 10
 C. 11
 D. 20
 E. more than 20

- 4.) Consider the following code fragment, where *i* is declared as an integer and *d* is type double:

```
switch (i) {
    case 10: d = 2.2; break;
    case 15:
    case 25: d = 7.7; break;
    default: d = 1.1; break;
}
```

This fragment is equivalent to which one of the following?

- A. **if** (*i* == 10) { *d* = 2.2; }
if (*i* == 15 || *i* == 25) { *d* = 7.7; }
d = 1.1;
- B. **if** (*i* == 10) { *d* = 2.2; }
else if (*i* == 15 || *i* == 25) { *d* = 7.7; }
else { *d* = 1.1; }
- C. **if** (*i* == 10) { *d* = 2.2; }
if (*i* >= 15 && *i* <= 25) { *d* = 7.7; }
d = 1.1;
- D. **if** (*i* == 10) { *d* = 2.2; }
else if (*i* >= 15 && *i* <= 25) { *d* = 7.7; }
else { *d* = 1.1; }
- E. **if** (*i* == 10) { *d* = 2.2; }
else if (*i* >= 15 || *i* <= 25) { *d* = 7.7; }
else { *d* = 1.1; }

- 5.) Consider the following code fragment:

```
String s1 = "one-fish";
String s2;
s2 = 2 + s1.substring(3);
System.out.println(s1 + "," + s2);
if (s1.equals(s2)) {
    System.out.println("blue fish");
}
else {
    System.out.println("red fish");
}
```

What is displayed when the code fragment is executed?

- A. one-fish,2-fish
red fish
- B. one-fish,2fish
red fish
- C. one-fish,2fish
blue fish
- D. one-fish,fishfish
blue fish
- E. one-fish,
2fish
blue fish

- 6.) Assume `month` is an integer variable representing the month number (i.e., 1 is January, 2 is February, etc.) and `day` is an integer variable representing the day of the month (i.e., 1 to 31). Which one of the following code fragments *best* implements code that displays the date in the form "month/day" except for the date October 31st, which displays "Happy Halloween" instead?

- A. `if (month != 10 && day != 31) {
 System.out.println(month + "/" + day);
}
else {
 System.out.println("Happy Halloween");
}`
- B. `if (month != 10) {
 if (day != 31) {
 System.out.println(month + "/" + day);
 }
 else {
 System.out.println("Happy Halloween");
 }
}`
- C. `if (month != 10) {
 System.out.println(month + "/" + day);
}
else if (day != 31) {
 System.out.println(month + "/" + day);
}
else {
 System.out.println("Happy Halloween");
}`
- D. `if (month == 10 && day == 31) {
 System.out.println("Happy Halloween");
}
else {
 System.out.println(month + "/" + day);
}`
- E. `if (month == 10) {
 if (day == 31) {
 System.out.println("Happy Halloween");
 }
 else {
 System.out.println(month + "/" + day);
 }
}`

- 7.) Consider the following code: `Random gen = new Random(11);`

Which one of the following statements about this code is *false*?

- A. `gen` references a `Random` object (random number generator) that has been seeded.
 B. The code `gen.nextInt(5) + 2` generates a random number in the range of 2 to 7 inclusive.
 C. The same sequence of numbers will be generated by `gen` each time the program using `gen` is run.
 D. `gen` is *not* an example of a primitive variable.
 E. `gen` can be used as an argument in a method call.

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

```

if (density > 100) {
    System.out.print("Class 1");
}
else if (density <= 200) {
    System.out.print("Class 2");
}
else {
    System.out.print("Class 3");
}

```

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

- A. "Class 1" is displayed if and only if `density` is greater than 200.
- B. "Class 2" is displayed if and only if `density` is less than or equal to 200.
- C. "Class 2" is displayed if and only if `density` is less than 100.
- D. "Class 3" is displayed if and only if `density` is equal to 100.
- E. "Class 3" will never be displayed.

9.) Consider the following code fragment (assume `stdin` is a properly constructed `Scanner` object and `answer` is a `String` object). Review the reference page if necessary.

```

do {
    // code here not shown ...
    System.out.print("Would you like to try again? ");
    answer = stdin.next();
} while (CONDITION);

```

Which one of the following replacements for CONDITION does *not* evaluate to true if a player enters "yes"?

- A. `answer == "yes"`
- B. `answer.equals("yes")`
- C. `answer.equalsIgnoreCase("yes")`
- D. `answer.charAt(0) == 'y'`
- E. `answer.charAt(0) == 'y' || answer.charAt(0) == 'Y'`

10.) Consider the following code fragments where `a` is an array of integers and `INIT` is an integer constant:

fragment 1

```

for (int i=INIT; i < a.length; i++) {
    System.out.print(a[i]);
}

```

fragment 2

```

int i = INIT;
while (i < a.length) {
    System.out.print(a[i]);
    i++;
}

```

Under which of the following circumstances will the two code fragments produce the same output?

- i.* `INIT` is equal to 0.
 - ii.* `INIT` is greater than 0 but less than the array's length.
 - iii.* `INIT` is greater than the array's length.
- A. *ii* only
 - B. *iii* only
 - C. *i* and *ii* only
 - D. *ii* and *iii* only
 - E. *i*, *ii*, and *iii*

11.) Consider the following incomplete method that is passed two character arrays *possibly of different lengths*:

```
public static boolean isDisjoint(char[] array1, char[] array2) {
    for (int i = 0; i < array1.length; i++) {
        for (int j = 0; EXPRESSION; j++) {
            if (array1[i] OPERATOR array2[j]) {
                return VALUE;
            }
        }
    }
    return true;
}
```

Which one of the following replacements for EXPRESSION, OPERATOR and VALUE, when used to complete the method above, results in true being returned if and only if array2 has none of the values in array1 (i.e., array1 and array2 have no values in common)?

	<u>EXPRESSION</u>	<u>OPERATOR</u>	<u>VALUE</u>
A.	<code>j < array1.length</code>	<code>!=</code>	<code>true</code>
B.	<code>j <= array1.length-1</code>	<code>==</code>	<code>true</code>
C.	<code>j < array2.length</code>	<code>!=</code>	<code>false</code>
D.	<code>j < array2.length</code>	<code>==</code>	<code>false</code>
E.	<code>j <= array2.length-1</code>	<code>!=</code>	<code>true</code>

12.) Assume the code fragment below was used in Program 1 to determine which phrase to use to describe the digital pet's facial expression based on its happiness score (assume score is an int variable and phrase is a String variable):

```
if (score >= 9) { phrase = "big smile"; }
else if (score >= 6) { phrase = "smile"; }
else if (score <= 3) { phrase = "frown"; }
else { phrase = "expressionless"; }
```

If this code was put in its own method, which method header below should be used?

- A. `public static void` getExpression()
- B. `public static char` getExpression(int score)
- C. `public static` String getExpression(int score)
- D. `public static` String getExpression(String word)
- E. `public static int` getExpression(String word)

13.) Consider the following method:

```
public static int mystery (int n) {
    int r = 0;
    while (n > 0) {
        r *= n;
        n--;
    }
    return r;
}
```

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

- A. It returns the numbers from 1 to n added together.
- B. It returns the numbers from 1 to n added together or 0 if n is less than 1.
- C. It returns the numbers from 1 to n multiplied together.
- D. It returns the numbers from 1 to n multiplied together or 0 if n is less than 1.
- E. It always returns 0.

14.) Consider the following class:

```
public class ArrayParameters {

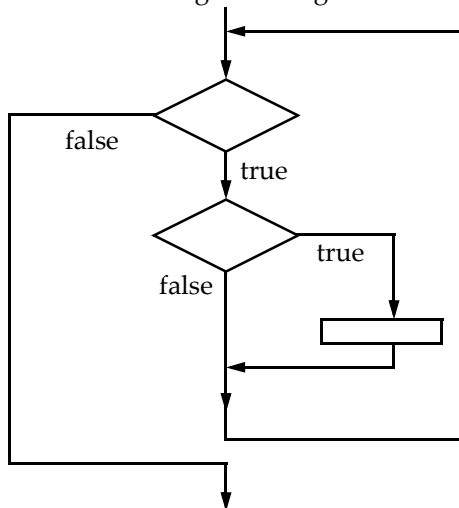
    public static void main(String[] args) {
        char[] a = {'a', 'e', 'i', 'o'};
        int n = 2;
        a = oops(a, n);
        for (int i = 0; i < a.length; i++) {
            System.out.print(a[i] + " ");
        }
    }

    public static char[] oops(char[] n, int a) {
        char[] c = {'r', 's', 't'};
        n[1] = c[0];
        c[a] = n[a];
        return c;
    }
}
```

Which one shows what the ArrayParameters program displays when executed?

- A. r s e
- B. r s i
- C. r s t
- D. a e i o
- E. a r i o

15.) Consider the following flow diagram:



This flow diagram represents which one of the following?

- A. an if statement with an if statement nested inside
- B. an if statement with a while loop nested inside
- C. a while loop with an if statement nested inside
- D. a while loop with an if-else statement nested inside
- E. a do-while loop with an if statement nested inside

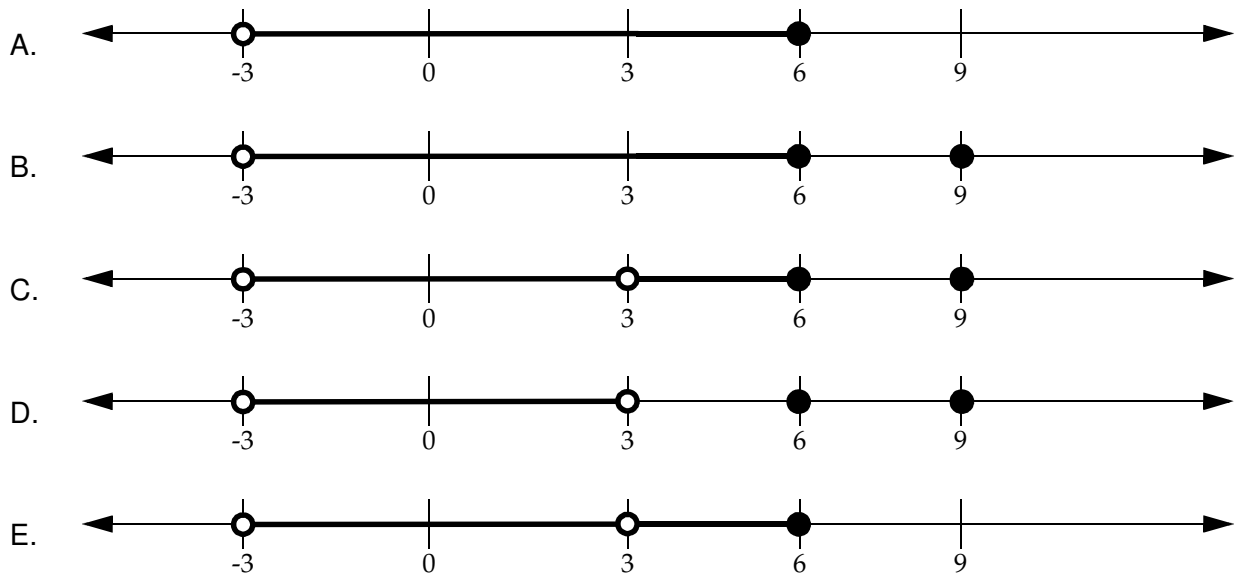
16.) Assume you are given three integer variables: `i1`, `i2` and `i3`. Which one of the following conditions *best* implements the idea of "three of a kind" (i.e., the variables have the same value)?

- A. `i1 == i2 == i3`
- B. `i1 && i2 && i3`
- C. `i1 == 11 && i2 == 11 && i3 == 11`
- D. `i1 == i2 && i2 == i3`
- E. `i1 == i2 && i2 == i3 && i1 == i3`

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

```
n > -3 && n <= 6 && n != 3 || 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.



18.) Consider the following formula for the surface area of a right circular cone:

$$a = \pi \times r^2 + \pi \times r \times \sqrt{h^2 + r^2}$$

Which one of the following *best* implements this formula in Java? Assume the variables `a`, `r`, and `h` used below have been declared to be type `double`.

- A. `a = 3.15 * r^2 + 3.15 * r * Math.sqrt(h^2 + r^2);`
- B. `a = Math.PI * r^2 + Math.PI * r * Math.sqrt(h^2 + r^2);`
- C. `a = 3.15 * r * r + 3.15 * r * Math.sqrt(h * h + r * r);`
- D. `a = Math.PI * Math.pow(2.0, r)`
`+ Math.PI * r * Math.sqrt(Math.pow(2.0, h) + Math.pow(2.0, r));`
- E. `a = Math.PI * Math.pow(r, 2.0)`
`+ Math.PI * r * Math.sqrt(Math.pow(h, 2.0) + Math.pow(r, 2.0));`

19.) Which one of the following statements about arrays in Java is *false*?

- A. An array variable's name is associated with a memory location that stores the address where its elements are found.
- B. An initial values list can be used to allocate and initialize the elements of an array.
- C. The length of an array can be increased by adding elements to its end.
- D. A run-time error will result when the array index is out of bounds for its array.
- E. When an array is passed to or returned from a method only its address is copied.

20.) Program 1 simulated 9 days in the life of a digital pet by calculating a pet's health, happiness and hunger based on the pet owner's actions. The code fragments below each go through 9 days displaying the day number for day. Which code fragment below accomplishes this using the *best* program structure?

A.

```
for (int i = 0; i < 9; i++) {
    if (i == 0) { System.out.println("Day 1:"); }
    else if (i == 1) { System.out.println("Day 2:"); }
    //assume similar else if's for days 3 - 8
    else if (i == 14) { System.out.println("Day 9:"); }
}
```

B.

```
for (int d = 1; d <= 9; d++) {
    switch (d) {
        case 1: System.out.println("Day 1:"); break;
        case 2: System.out.println("Day 2:"); break;
        //assume similar cases for days 3 - 8
        case 9: System.out.println("Day 9:"); break;
    }
}
```

C.

```
int d = 1;
for (int i = 0; i < 9; i++) {
    System.out.println("Day " + d + ":");
    d++;
}
```

D.

```
for (int i = 0; i < 9; i++) {
    System.out.println("Day " + (i + 1) + ":");
}
```

E.

```
for (int d = 1; d <= 9; d++) {
    System.out.println("Day " + d + ":");
}
```

21.) Consider the following expression containing operators (identified by the numbers above them) and where a, t, z and u are variables of type double:

$$a = \overset{1}{(t * \overset{2}{z} / \overset{3}{2.0})} - (\overset{4}{t} * \overset{5}{t} + \overset{6}{3.7} / \overset{7}{u})$$

Which one lists the order that the operators are executed when the expression is evaluated? The first operator to be executed is operator 2, so its listed first. Review the reference page if necessary.

- A. 2, 3, 5, 7, 6, 1, 4
- B. 2, 3, 5, 7, 6, 4, 1
- C. 2, 3, 7, 5, 6, 4, 1
- D. 2, 5, 3, 7, 6, 4, 1
- E. 2, 5, 3, 7, 4, 6, 1

22.) Consider the following code fragment where `a` is an array of integers and `stdIn` is a properly initialized Scanner:

```

boolean done = false;
while (!done) {
    //location A
    System.out.print("Enter an integer: ");
    int x = stdIn.nextInt();
    for (int i = 0; i < a.length; i++){
        //location B
        if (a[i] == x) {
            //location C
            done = true;
        }
    }
    //location D
}
//location E

```

At which of the locations labeled above can the code `System.out.print(x);` be added without causing a compile-time error?

- A. only locations A and E
- B. only locations B and C
- C. only locations B and D
- D. only locations B, C and D
- E. only locations B, C, D and E

23.) Consider the following program:

```

public class Parameters {

    public static void main(String[] args) {
        int x = 1, y = 4, z = 7;
        z += flip(x, y);
        System.out.println(x + y + z);
    }

    public static int flip(int y, int x) {
        y++;
        x--;
        return flop(x, y);
    }

    public static int flop(int x, int y) {
        return x - y;
    }
}

```

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

- A. 6
- B. 7
- C. 13
- D. 147
- E. 148

24.) Consider the following program:

```
public class Methods {

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

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

```
int count = 1;
boolean done = false;

do {

    if (count > 4) {
        done = true;
    }
    else if (count % 2 == 1) {
        count += 3;
    }
    else {
        count--;
    }

    System.out.print(count + ", ");

} while (!done);
```

- A. 4,
- B. 4, 4,
- C. 4, 3, 6,
- D. 4, 3, 6, 6,
- E. The loop displays numbers continuously since it's an infinite loop.