Computer Sciences 302
Midterm Exam 1, 20%

Thursday 10/21, 2010

Print last name: ______________________________, first: ____________

Signature: ______________________________ CS login: ____________

Circle Your Lecture	**Lec 1**	**Lec 2**	**Lec 3**	**Lec 4**	**Lec 5**	**Lec 7**	**Lec 8**	**Lec 9**	**Lec 10**
	Skrentny	Skrentny	Dalibor	Dalibor	Finn	Dan	Finn	Alex	Alex

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 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 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 23 question each worth 3 points with a maximum score of 66 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

lower	`= += -=`
	`*= /= %=`
		assignment and
		compound assignments

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

```
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.
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 beginIndx, int endIndx)
                        //Returns a new string that is a substring of this string
                        //starting at beginIndx up to but not including endIndx.
```

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

```
Random()                //Creates a new random number generator.
int nextInt()           //Returns the next pseudorandom integer value.
int nextInt(int n)      //Returns the next pseudorandom 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.
String next()           //Returns the next "word" of input.
double nextDouble()     //Returns the next input as a double.
int    nextInt()        //Returns the next input as an integer.
String nextLine()       //Returns the next input line as a String.
```

1.) Consider the following program:

```
public class Parameters {

   public static void main(String[] args) {
      int a = 1, b = 2, c = 3;
      a = messUp(b, a);
      System.out.println(a + b + c);
   }

   public static int messUp(int a, int b) {
      int c = 4;
      a++;
      b--;
      return a + c;
   }
}
```

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

A. `11`
B. `12`
C. `13`
D. `123`
E. `723`

2.) Consider the following code fragment:

```
int a = 11, b = 22, c = 33;

c = a;
a = b;
b = c;
```

Which one of the following shows what the values of the variables would be after the code executes?

```
   a    b    c
A. 22   11   11
B. 22   11   33
C. 22   22   33
D. 33   22   11
E. 33   33   33
```

3.) What is the output of the following code fragment?

```
int n = 1;
while (n != 7) {
   System.out.print(n);
   if (n % 2 != 0) {
      n--;
   }
   n += 2;
}
```

A. `135`
B. `1357`
C. `122222222`... (the code results in an infinite loop)
D. `1246`
E. `124681012`... (the code results in an infinite loop)

4.) Which one of the following *best* describes the line of code below:

```
final double RATE = 3.5;
```

A. RATE is declared as the number 3.5.
B. RATE is declared as a constant equal to 3.5.
C. A reference variable named RATE is declared and is assigned 3.5.
D. A double variable named RATE is declared and is given an initial value of 3.5.
E. A variable is declared with the final name RATE and is given an initial value of 3.5.

5.) Consider the following method where s1 and s2 are properly initialized String objects:

```
public static boolean doIt(String s1, String s2) {
   int i1 = 0, i2 = 0;
   while (i1 < s1.length()) {
      if (s1.charAt(i1) == s2.charAt(i2)) {
         i1++;
         i2 = 0;
      }
      else if (i2 < s2.length() - 1){
         i2++;
      }
      else {
         return false;
      }
   }
   return true;
}
```

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

A. It always returns true.
B. It returns true if and only if s1 and s2 begin with the same character.
C. It returns true if and only if s2 begins with s1 (e.g., s1 has "to" and s2 has "town").
D. It returns true if and only if s1 and s2 are the same strings (e.g., s1 has "leap" and s2 has "leap").
E. It returns true if and only if all of the characters in s1 are found somewhere in s2.

6.) Consider the following code fragment:

```
for (int j = 1; j <= 3; j++) {
   System.out.print(j + ":");
   for (int k = j; k >= 0; k--) {
      System.out.print("*");
   }
   System.out.println();
}
```

Which one of the following shows what is displayed by this code fragment?

A. `1:2:*3:**`

B. `1:*2:**3:***`

C. `1:**2:***3:****`

D.
```
1:*
2:**
3:***
```

E.
```
1:**
2:***
3:****
```

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

```
for (int i = 1; i <= 6; i++) {
   switch (i) {
   case 1: System.out.print("A"); break;
   case 2: System.out.print("B");
   case 4:
   case 5: System.out.print("C"); break;
   case 6: System.out.print("D"); break;
   default: System.out.println();
   }
}
```

A. `ABCCD`

B. `ABCCCD`

C.
```
AB
CD
```

D.
```
AB
CCD
```

E.
```
ABC
CCD
```

8.) Given the following variable declarations and comments describing their use:

```
int hour;       // 1 - 12, hour of the day
int minute;     // 0 - 59, minute of the hour
boolean pm;     // true if it is PM, false if it is AM
```

Which one of the choices below describes the following condition:

```
(hour == 7 && minute >= 5 || hour == 8 || hour == 9 && minute <= 5) && pm
```

A. from 7:05 AM to 9:05 AM
B. from 7:05 PM to 9:05 PM
C. either 7:05 PM, 8:00 PM, or 9:05 PM
D. either 7:05 to 7:59 AM, or 8:00 AM, or 9:00 to 9:05 AM
E. either 7:05 to 7:59 PM, or 8:00 PM, or 9:00 to 9:05 PM

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

```
int n = 1;
int sum = 0;
if (n <= 3) {
   sum = sum + n;
   n++;
}
System.out.println(sum);
```

A. `0`
B. `1`
C. `3`
D. `4`
E. `6`

10.) Program 1 was a game that simulated running a lemonade stand for 14 days. Consider the code fragments below that each loop 14 times displaying the day number each time. Which code fragment below accomplishes this using the *best* program structure?

A.
```
for (int i = 0; i < 14; 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 - 13
   else if (i == 13) { System.out.println("Day 14:"); }
}
```

B.
```
for (int d = 1; d <= 14; 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 - 13
      case 14: System.out.println("Day 14:"); break;
   }
}
```

C.
```
for (int d = 1; d <= 14; d++) {
   System.out.println("Day " + d + ":");
}
```

D.
```
int d = 1;
for (int i = 0; i < 14; i++) {
   System.out.println("Day " + d + ":");
   d++;
}
```

E.
```
int d = 0;
while (d < 14) {
   System.out.println("Day " + (d + 1) + ":");
   d++;
}
```

11.) Consider the following two code fragments where `score` is an integer variable:

fragment 1
```
if (score > 75) {
   System.out.print("pass");
}
else if (score < 75) {
   System.out.print("fail");
}
```

fragment 2
```
if (score > 75) {
   System.out.print("pass");
}
else {
   System.out.print("fail");
}
```

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

i. `score` is less than 75
ii. `score` is 75
iii. `score` is greater than 75

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

12.) Assume you are writing a method to return the index of the last match of a search character in a list that is a partially-filled array of characters. Which one of the following should be used as the method header?

A. `public int findLastMatchIndex(char list, char search)`
B. `public int findLastMatchIndex(char[] list, int numUsed, char search)`
C. `public static int findLastMatchIndex(char[] list, char search)`
D. `public static int[] findLastMatchIndex(char list, int numUsed, char search)`
E. `public static int findLastMatchIndex(char[] list, int numUsed, char search)`

13.) Consider the following code fragment, where `i` is an integer variable and `d` is a double:

```
if (i == 1) {
   d = 2.2;
}
else if ((i == 2) || (i == 3)) {
   d = 7.7;
}
else {
   d = 1.1;
}
```

This code fragment is equivalent to which one of the following?

A.
```
switch (i) {
case 1: d = 2.2;
case 2:
case 3: d = 7.7;
case >3: d = 1.1;
}
```

B.
```
switch (i) {
case 1: d = 2.2; break;
case 2:
case 3: d = 7.7; break;
default: d = 1.1;
}
```

C.
```
switch (i) {
case 1: d = 2.2; break;
case 2:
case 3: d = 7.7; break;
case >3: d = 1.1;
}
```

D.
```
switch (i) {
case 1: d = 2.2; break;
case 2: break;
case 3: d = 7.7; break;
case >3: d = 1.1; break;
}
```

E.
```
switch (i) {
case 1: d = 2.2; break;
case 2: break;
case 3: d = 7.7; break;
default: d = 1.1; break;
}
```

14.) 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

15.) Program 1 required a (pseudo) random integer in the range from 15 to 25 (inclusive) be generated for the price of sugar. Assume the variable `ranGen` references a properly initialized `Random` class object and `sugarPrice` is an integer variable. Which code fragment below ***best*** accomplishes this requirement?

A. `sugarPrice = ranGen.nextInt(11) + 15;`

B. `sugarPrice = ranGen.nextInt(16) + 10;`

C. `sugarPrice = ranGen.nextInt(26) + 15;`

D.
```
sugarPrice = ranGen.nextInt(26);
if (sugarPrice < 15) {
   sugarPrice += 15;
}
```

E.
```
sugarPrice = 11;
while (sugarPrice < 15 || sugarPrice > 25) {
   sugarPrice = ranGen.nextInt();
}
```

16.) Of the numbered operators in the expression below, which one is evaluated first? Assume `a`, `b` and `c` are integer variables. Review the reference page if needed.

```
        1     2    3    4       5
a != 11 && (a < (c + 22)/3 || b == 100)
```

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

17.) Consider the following code fragment (review the reference page for these method calls):

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

What is printed when the code fragment is executed?

A.
```
one-fish, 2fish
red fish
```

B.
```
one-fish, 2fish
blue fish
```

C.
```
one-fish, 2-fish
red fish
```

D.
```
one-fish, 2-fish
blue fish
```

E.
```
one-fish,
2-fishblue fish
```

18.) Which one of the following statements about method calls is true?

A. A method call must include a list of the types of its parameter values (i.e., arguments).
B. A method call must have the name of the class where the method is defined.
C. A method call that passes an array must have empty brackets (i.e., `[]`) after the method's name.
D. A method called without any parameter values (i.e., arguments) must have empty parentheses (i.e., `()`) after the method's name.
E. A method call can have more parameter values (i.e., arguments) listed than the number of parameter variables specified in that method's header.

19.) Consider the following incomplete method:

```
public static void rotateToLeft(double[] data) {
   double d = data[0];
   for (int i = 0; i < data.length - VALUE; i++) {
      data[i] = data[EXPRESSION];
   }
   STATEMENT
}
```

Which one of the following replacements for `VALUE`, `EXPRESSION` and `STATEMENT`, when used to complete the method above, results in rotating the values in the array `data` to the left by one element? For example, if the array originally had 1.1, 2.2, 3.3, it would have 2.2, 3.3, 1.1 after the method executes.

	VALUE	EXPRESSION	STATEMENT
A.	1	i + 1	return;
B.	2	i - 1	return;
C.	1	i + 1	data[data.length - 1] = d;
D.	1	i - 1	data[data.length + 1] = d;
E.	2	i + 1	data[data.length - 1] = d;

20.) Assume an array of integers from 0 - 100 inclusive, named `percents`, has been filled with valid values in no particular order. Which one of the following code fragments *best* implements the code so that finds the value of the second smallest percentage in the array.?

A.
```
int secondSmallest = percents[1];
```

B.
```
int smallest = 0, secondSmallest = 0;
for (int i=0; i < percents.length; i++) {
   if (percents[i] > smallest) {
      secondSmallest = smallest;
      smallest = percents[i];
   }
}
```

C.
```
int smallest = 0, secondSmallest = 0;
for (int i=0; i < percents.length; i++) {
   if (percents[i] > smallest) {
      secondSmallest = smallest;
      smallest = percents[i];
   }
   else if (percents[i] > secondSmallest) {
      secondSmallest = percents[i];
   }
}
```

D.
```
int smallest = 101, secondSmallest = 101;
for (int i=0; i < percents.length; i++) {
   if (percents[i] < smallest) {
      secondSmallest = smallest;
      smallest = percents[i];
   }
}
```

E.
```
int smallest = 101, secondSmallest = 101;
for (int i=0; i < percents.length; i++) {
   if (percents[i] < smallest) {
      secondSmallest = smallest;
      smallest = percents[i];
   }
   else if (percents[i] < secondSmallest) {
      secondSmallest = percents[i];
   }
}
```

21.) Consider the following formula for the distance between points (x_1,y_1) and (x_2,y_2):

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

Which one of the following is a correct Java implementation of this formula. Assume the variables below `d`, `x1`, `y1`, `x2`, and `y2` have been declared to be type `double`.

```
A. d = Math.sqrt(Math.pow(x2 - x1, y2 - y1, 2.0));
B. d = Math.sqrt(Math.pow((x2 - x1) + (y2 - y1), 2.0));
C. d = Math.sqrt(Math.pow(x2 - x1, 2), Math.pow(y2 - y1, 2));
D. d = Math.sqrt(Math.pow(x2 - x1, 2.0) + Math.pow(y2 - y1, 2.0));
E. d = Math.sqrt(Math.pow((x2 - x1), 2.0), Math.pow((x2 - x1), 2.0));
```

22.) Consider the following method:

```
public static void patternFill(char[][] board) {
   boolean x = true;
   for (int r = 0; r < board.length; r++) {
      for (int c = 0; c < board[r].length; c++) {
         if (x) {
            board[r][c] = 'X';
            x = false;
         }
         else {
            board[r][c] = 'O';
            x = true;
         }
      }
   }
}
```

Which one of the following correctly shows the pattern in `board` after it has been filled by this method?

A. A 4 by 4 board would be filled with this pattern:

X	O	X	O
O	X	O	X
X	O	X	O
O	X	O	X

B. A 3 by 4 board be filled with this pattern:

X	O	X	O
O	X	O	X
X	O	X	O

C. A 3 by 3 board would be filled with this pattern:

X	O	X
O	X	O
X	O	X

D. A 4 by 4 board would be filled with this pattern:

O	X	O	X
X	O	X	O
O	X	O	X
X	O	X	O

E. A 3 by 4 board would be filled with this pattern:

O	X	O	X
X	O	X	O
O	X	O	X

23.) Consider the following program:

```
public class ArrayParameters {

   public static void main(String[] args) {
      int[] array = {9, 8, 7, 6};
      array[2] = mystery(array);
      for (int i = 0; i < array.length; i++) {
         System.out.print(array[i] + " ");
      }
   }
   public static int mystery(int[] a) {
      a[1] = -3;
      a = new int[4];
      a[3] = -2;
      return a[2];
   }
}
```

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

A. `9 -3 0 6`
B. `9 -3 7 -2`
C. `9 -3 7 6`
D. `9 8 0 6`
E. `9 8 7 6`