My Bb Home My F Darek Konopka

Week 10 **Review Test Submission: Sample Final**

Review Test Submission: Sample Final

User	Darek Konopka
Course	CS-101-01: Computing & Algorithms I - SPRING MWR 8:00-10:05
Test	Sample Final
Started	6/11/20 7:50 AM
Submitted	6/11/20 9:49 AM
Due Date	6/11/20 10:30 AM
Status	Needs Grading
Attempt Score	Grade not available.
Time Elapsed	1 hour, 59 minutes out of 2 hours
Instructions	Please read the following carefully: • This exam is timed. You will have two hours to complete the exam once you start it. • This exam can be saved and resumed at any point until time has expired. **The timer will continue to run if you leave the exam.** • Once the time limit is reached, this exam will submit automatically. • You only have one attempt at this exam. Please take your time to complete it, and read each question carefully before answering. Good luck!
Results Displayed	All Answers, Submitted Answers, Correct Answers

Question 1

Which of the following statements is correct to assign character 5 to c?

Selected Answers: ✓ `char c = '5';`

Answers:

✓ `char c = '5';`

`char c = 5;`

`char c = "5";`

`char c = (char)5;`

Question 2

Consider the following code:

```
Scanner input = new Scanner( System.in );
int number = input.nextInt();
if ( number <= 0 )
   System.out.println( number );
System.out.println( number );
```

Which of the following is/are correct? Select all that apply.

Selected Answers: ✓ The value of `number` is printed out at least once.

✓ The value of `number` is printed out twice if its value is 0.

Answers: ✓ The value of `number` is printed out at least once.

✓ The value of `number` is printed out twice if its value is 0.

The value of `number` is printed out twice if its value is positive.

✓ The value of `number` is printed out twice if its value is nonnegative.

Question 3

Which of the following is/are true? Select all that apply.

Selected Answers: ✓ A non-static method in a class can access any other method in the same class.

✓ If a method in a subclass overrides a public method in the superclass, then a client can use a superclass object reference to call the overriding method.

Answers: ✓ A method's parameters cannot be accessed by any other method.

✓ A non-static method in a class can access any other method in the same class.

If a method in a subclass overloads a public method in the superclass, then the superclass version of the method cannot be called by a subclass client using a subclass object reference.

✓ If a method in a subclass overrides a public method in the superclass, then a client can use a superclass object reference to call the overriding method.

Question 4

Which Java keyword do we use in the constructor of a class inheriting from another class if we want to call the constructor of the inherited class?

Selected Answer: super()

Correct Answer: ✓ super

Question 5

Suppose we have an *Auto* class correctly coded, then given the following statement:

```
Auto [ ] cars = new Auto[2];
```

what is the value of `cars[1]`?

Selected Answer: In that first statement we define the length of the object to be 0-2
and since cars[1], is not set, we would use the default values .
Openning the Auto course code, we can see that the default is
Auto(unknown, 0, 0) where if we use the .toString() method it would output
"Model: " + unknown + ;
"miles driven: " + 0 +;
"gallons of gas: " + 0;

Correct Answer: ✓ null

Question 6

Given the following statement:

```
boolean [ ][ ] booleanArray = new boolean [10][15];
```

(a) What will be the value of `booleanArray[8].length`?

(b) What will be the value of `booleanArray[1][1]`?

Selected Answer: Here we deifne the size of the array
so the size of `booleanArray[8]` would be 15
and since we didnt define anything in specific except the size, then all the values in the array at this moment will be 0
but since this is an array of booleans, all 0's are equal to false
therefore `booleanArray[1][1] = false`

Correct Answer: (a) 15
✓ (b) false

Question 7

If `d1` and `d2` are two *SimpleDate* object references, then when will `( d1 == d2 )` evaluate to `true` and when will `d1.equals(d2)` evaluate to `true`?

Selected Answer: `d1 == d2`
will evaluate to true if they are they same exact object, not only if they have the same values, but if they are the exact same object
and `d1.equals(d2)`
will return true if the values of d1 match that of d2

Correct Answer: ✓
(d1 == d2) will evaluate to true when d1 and d2 are pointing to the same object, that is, the same memory location.

d1.equals(d2) will evaluate to true if the data values in d1 object are equivalent to the data values in d2 object.

Question 8

For a sentinel-controlled loop, if we know that the loop termination condition is:

```
input == 0 || input > 35
```

What should be the loop continuation condition? Please apply the DeMorgan's Laws to simplify your expression so that your final expression does not contain any parentheses.

Selected Answer:

```
!(input == 0 || input > 35)
(input!=0 && input <= 35)
```

Correct Answer: ✓

The loop continuation condition is: `!( input == 0 || input > 35 )`

Apply the DeMorgan's Law, we have: `!( input == 0 ) && !( input > 35 )`

Remove the parentheses: `input != 0 && input <= 35`

Question 9

Identify the error in the following statements:

```
int a = Math.ceil( 12.5 );
System.out.println( a );
```

Selected Answer:

When using the Math.ceil function, that returns a double
in order to fix this we either convert ceil to an int by typing

`int a = (int) Math.ceil( 12.5 );`

or we can make our varible a double

`double a = Math.ceil( 12.5 );`

but the first option is most likely what we would use since the code wanted us to keep it as an int

Correct Answer: ✓

The return type of the method `ceil` is `double`, that is, `Math.ceil( 12.5 )` will return a `double` and it cannot be assigned to an `int` variable.

Question 10

Consider the following *Print* class.

```
public class Print
{
   private String s;
   public Print ( )
   { }
   public Print ( String s )
   {
      this.s = s;
   }
   private void print ( )
   {
      System.out.println( s );
   }
   @Override
   public String toString ( )
   {
      return s;
   }
}
```

(a) We are adding the following statements to a client class of *Print* as follows; what is the error?

```
Print a = new Print ( "Hello!" );
a.print( );
```

(b) We want to create a subclass of *Print*, named *PrintNumber*, and we are adding the following members to it; what is the error?

```
private int num;
public PrintNumber ( String s, int num )
{
   super.Print( s );
   this.num = num;
}
```

Selected Answer:

a.) The client class and the Print class should not be named the same, this will cause an error
name your client class something like PrintClient instead so you know it is the client class of the print class

b.) `super.Print( s );`

what we should do instead is just type

super (s);

and this will extend the variable we need

Correct Answer:

✓
(a) The print( ) method is declared private, so it cannot be called by the client.
(b) To call a constructor of the superclass, the keyword super should be used as follows: super(s);

Question 11

How many times will the print statement be executed? Explain your answer.

```
for (int i = 15; i > 10; i--)
   for (int j = 1; j < i; j++)
      System.out.print( i + j + " " );
```

Selected Answer:

This should print 60 times
for the first for loop we start at 15 and go down until we reach 10, so the first for loop will run the second 5 times
then the second loop we count up until our i -1 so:
i = 15 j will count up to 14
i = 14 j will count up to 13
and so on until i = 10
then we do 14+13+12+11+10 and we get **60**

Correct Answer: ✓

There are five iterations in the outer for loop.
In the 1st iteration, i = 15, and there are 14 iterations in the inner for loop (j starts from 1 to 14, incremented by 1 in each iteration).
In the 2nd iteration, i = 14, and there are 13 iterations in the inner for loop (j starts from 1 to 13, incremented by 1 in each iteration).

In the 3rd iteration, i = 13, and there are 12 iterations in the inner for loop (j starts from 1 to 12, incremented by 1 in each iteration).

In the 4th iteration, i = 12, and there are 11 iterations in the inner for loop (j starts from 1 to 11, incremented by 1 in each iteration).

In the 5th iteration, i = 11, and there are 10 iterations in the inner for loop (j starts from 1 to 10, incremented by 1 in each iteration).

Thus in total, there are 14 + 13 + 12 + 11 + 10 = 60 iterations. So, the print statement will be executed **60 times** as it will be executed once in each iteration.

Question 12

Explain what the following code sequence is doing.

```
for ( int i = 3; i < numbers.length - 1; i++ )
{
   int m = 3;
   for ( int j = 4; j < numbers.length - i + 3; j++ )
   {
      if ( numbers[j] < numbers[m] )
         m = j;
   }
   int temp = numbers[m];
   numbers[m] = numbers[numbers.length - i + 2];
   numbers[numbers.length - i + 2] = temp;
}
```

Consider an int array `numbers` with initial values {11, 53, 3, 21, 15, 8, 45, 2, 32}. Write the values of `numbers` after each iteration of the outer for loop of the above code.

Selected Answer:

The purpose of the is to sort the array of numbers in ascending order using a selection sort however it is not done correctly so it is kind of weird

if we have initial values {11, 53, 3, 21, 15, 8, 45, 2, 32} for numbers then after each itteration here is what we will have

iteration1: {11, 53, 2, 21, 15, 8, 45, 32, 2}
iteration 2: {11, 53, 2, 21, 15, 32, 45, 8, 2}
iteration 3: {11, 53, 2, 21, 45, 32, 15, 8, 2}
iteration 4: {11, 53, 2, 32, 45, 21, 15, 8, 2}
iteration 5: {11, 53, 2, 45, 32, 21, 15, 8, 2}

Correct Answer: ✓

This code is sorting the subarray of `numbers` from index 3 (the fourth element) to the last element of the array.

When the value of `numbers` is {11, 53, 3, 21, 15, 8, 45, 2, 32}, the outer for loop will have 5 iterations.

At the end of the 1st iteration, the array becomes: 11, 53, 3, 21, 15, 8, 45, 32, 2
At the end of the 2nd iteration, the array becomes: 11, 53, 3, 21, 15, 32, 45, 8, 2
At the end of the 3rd iteration, the array becomes: 11, 53, 3, 21, 45, 32, 15, 8, 2
At the end of the 4th iteration, the array becomes: 11, 53, 3, 32, 45, 21, 15, 8, 2
At the end of the 5th iteration, the array becomes: 11, 53, 3, 45, 32, 21, 15, 8, 2

Question 13

Write a program using nested loops to produce the following output:

```
1 2 4 8 16
1 2 4 8
1 2 4
1 2
1
```

You may use any combination of nested loops we've learned in this course. There will be no more than three print statements in your program.

Selected Answer: NestedLoop.java

Question 14

Write a program to

(1) Create a two-dimensional array of the sales tax rates for 5 states over 10 years, with dimension 1 representing the state and dimension 2 representing the year.

(2) For each element of the array, generate a random number between 0 and 6, inclusive, and divide it by 100. So, the sales tax rates will be between 0 and 0.06.

(3) Output the array with each row starting from a new line and columns being separated by a space.

(4) Find the state that has the lowest tax rate in the last (10th) year, and output its index.

(5) Find the state that has the biggest average tax rate over the 10 years, and output its index.

Selected Answer: TwoDFun.java

Question 15

We have a *BankAccount* class *BankAccount.java* and a subclass *CheckingAccount.java* already written. Write a client program to complete the following tasks:

(1) Create an array named `accounts` with 5 *BankAccount* object references.

(2) For each object reference, assign a *CheckingAccount* object to it by prompting the user for the values of `balance` and `monthlyFee`. You may assume that the user will always enter a `double` for each value.

(3) Sort the array `accounts` in descending order using `monthlyFee` as the sort key. Use any of the three sorting algorithms, selection sort, insertion sort, or bubble sort, to perform the sorting.

(4) Output the sorted array by displaying both the `balance` and `monthlyFee` values for each element.

Selected Answer: Accounts.java

Thursday, June 18, 2020 12:12:58 AM EDT

← **OK**