FINAL EXAM  Introduction to Computer Science UA-CCI- ICSI 201--Fall13

This is a closed book and note examination, except for one 8 1/2 x 11 inch paper sheet of notes, both sides. There is no interpersonal or other communication or discussion (except with the prof. or proctors) or sharing of information. Cell phones or any electronic devices, other than a calculator or a pre-approved dedicated language translator are strictly forbidden! If you need to leave the room, bring all your papers to the proctor; only one person will be permitted to leave at a time. Failure to abide by any of these terms may result in a grade zero for the exam.

16 multiple choice questions: 0 pts for omitted or wrong answers, 4 pts for correct answer, 64 points max.

One multi-part 35 point Java programming question. The programming question will be graded on the basis of CLARITY and how well the form of expression that indicates you know what you are doing. Pencil use is preferred and pencils with erasers are available for you to borrow.

1. Write your name clearly on the front of the Scantron sheet.
2. ON THE BACK of the Scantron, print your last name first AND BUBBLE IN THE BOXES.
3. ON THE BACK of the Scantron, Print your UAlbany 9 decimal digit ID number AND BUBBLE IN THE BOXES. IF YOU DON'T KNOW (or are unsure of) your ID number, take a break during the exam and have the prof. or TA look up and write the number for you.
4. Don't bother with gender, birthday, etc. But if you omit writing and bubbling in YOUR NAME AND YOUR 9-digit ID, we will do it AND DEDUCT 5 points from your exam score!

4. RIP OFF this front page of this bundle of questions. WRITE YOUR NAME AND NET-ID CLEARLY ON THE BACK, and use the back to write the Java programming exam assignment.

5. PLEASE TAKE HOME the question bundle, but turn in 2 paper sheets: The scantron AND the paper where you wrote your programming question solution!

6. Continue your Java coding onto the space below if you need to. Also, fresh paper is available on request.
import java.util.Scanner;
public class ___________________ {
    public static void main(String[] a) {
        //Write name&NetID so we can record your programming grade!
        System.out.println("Author_____________________________");
        System.out.println("NetID:_________________");
        int sum, largest, whereIsLargest, nTimesAppears; //REMEMBER TO INITIALIZE ALL
        //VARIABLES! (You may use shorter names as long as they are not misleading.

        for (int i = 0; i < 10; i = i + 1 )
        {

        }

        //continue your work on page 1 if necessary.
        System.out.println("The sum is " + sum);
        System.out.println("The largest is " + largest);
        System.out.println("One index where the largest is located is " +
        whereIsLargest );
        System.out.println("The largest was input "+ nTimesAppears +" times.");
    } //end of the body of method main.
} //end of class definition
(20-early 21st Century electronic) computers follow the instructions coded in their programs:
(A.) Exactly in the order you have written them in the program, as modified by the method call and return, while/for loops, and if/else conditionals which you have also written.
(B.) In the order that makes sense to solve the problem correctly.
(C.) Randomly.

When the computer runs the code

```java
Picture picRef = new Picture( FileChooser.pickAPath());
```

which of the three actions A., B., C. happens first?
(A.) Method `FileChooser.pickAPath()` is called and it returns a Java String (reference).
(B.) A new `Picture` object is built according to the blueprint that Profs. Guzdial and Ericson wrote into their `Picture.java` you got with the book classes.
(C.) The address of the `Picture` object is copied and becomes the value in the variable named `picRef`.

Within the above code again, which Java operation means build a new `Picture` object?
(A.) `FileChooser.pickAPath()`
(B.) `new Picture( ... )`
(C.) `=`
(D.) `Picture picRef`

Within the above code again, which Java operation means copy the address of the new `Picture` object?
(A) `FileChooser.pickAPath()`
(B) `new Picture( ... )`
(C.) `=`
(D.) `Picture picRef`

Questions about how actual Java code controls what the computer does.
Suppose the user successfully picks the three images `One.jpg`, `Two.jpg` and `Three.jpg` in that order when he or she runs the program BELOW. Which two images will he or she actually see?

```java
public class Fall13FinalAppOne
{
    public static void main(String[]a)
    {
        FileChooser.pickMediaPath();
        Picture Sally;
        Picture Charlie;
        Sally = new Picture( FileChooser.pickAFile() );
        Charlie = new Picture( FileChooser.pickAFile() );
        Sally.explore();
        Charlie.explore();
    }
}
```

(a.) `One.jpg` and `Two.jpg`
(b.) `One.jpg` and `Three.jpg`
(c.) `Two.jpg` and `Three.jpg`
(d.) cannot tell from the given problem information.
Here are three of the first things the computer does when it runs a different application below, in the 1,2,3 order in which it did them.

1. The application below first pops up a window for you to choose a directory (i.e., folder) where you have some .jpg digital image files.
2. It then pops up another file-chooser window for you to pick one .jpg image file.
3. Assume you pick a small 20x20 (width x height) pixel image file.

After you pick that small 20x20 image file, HOW MANY MORE TIMES DOES a file-chooser window pop up? Assume that whenever you pick a file, the picking and making of a Picture is successful. Remember this question detail: Do NOT count the window to pick the media path and do NOT count first file-chooser window popping up!

```java
public class Fall13FinalAppTwo {
    public static void main(String[] a) {
        FileChooser.pickMediaPath();
        Picture pictRefOne;
        pictRefOne = new Picture( FileChooser.pickAFile() );
        if( pictRefOne.getWidth() > 30 ) {
            System.out.println("Please try again.");
            pictRefOne = new Picture( FileChooser.pickAFile() );
        }
        Picture pictRefTwo = new Picture( FileChooser.pickAFile() );
        pictRefOne.explore(); //first Picture explored.
        pictRefTwo.explore(); //second Picture explored.
    }
}
```

(A.) 0
(B.) 1
(C.) 2
(D.) 3
(E.) cannot tell from the given problem information.

To introduce the concept, we showed the Mad Ph.D. video and used the plain English terms "build" and "make" for what the computer does when it runs the Java code `new Happy_House( )`; What are the technical terms in Java technology that mean the same thing as "make" or "build" when that action is done by Java's `new` operation?

(A.) construct or instantiate
(B.) define or construct
(C.) initialize or declare
(D.) newing or declaring
(E.) instantiate or compile
Sometimes it's OK to write different similar looking variations of code to achieve the same purpose, but sometimes that is **NOT** OK. Which pair of similar looking code examples are NOT OK to interchange?

(A) `if( A == B) { sames++; }` and `if( B == A ) { sames = sames + 1; }`
(B) `if ( A >= B ) { bigger = A; }` and `if( B <= A ) { bigger = A; }`
(C) `I=I+1;` and `I = I + 1;`
(D) `I=I+1;` and `I+1=I;`
(E) `while( I < A.length ) { System.out.println( A[I] ); }` and `while( I <= A.length-1 ) { System.out.println( A[I] ); }`

Which explanation of the Java expression `A[i+1]` best explains how the value and the location of a Java array element are denoted in Java syntax? Suppose the array was made (instantiated) by `int[] A = new int[10];`

(A) In `A[i+1]`, `A` is the array name, `i` is the element location, and `i+1` is the element's value.
(B) In `A[i+1]`, `i` is the array name, `A` is the location, and `i+1` is the element's value.
(C) In `A[i+1]`, `A` is the array name, `i` is the element location, and `A[i+1]` is the element's value.
(D) In `A[i+1]`, `A` is the array name, `i+1` is the element location, and `A[i+1]` is the element's value.
(E) In `A[i+1]`, `A` is the array name, `A[i]` is the element location, and `A[i+1]` is the element's value.

When the computer finds the maximum of the heights of the 6 Pictures as it should in Project 6, what should it do to decide whether or not to replace the number previously found to be the maximum so far?

(A) In the ( ... ) of a `while` or `for` statement, compare the previous Picture's height to the current Picture's height.
(B) In the ( ... ) of a `while` or `for` statement, compare the maximum height found so far to the current Picture's height.
(C) In the ( ... ) of an `if` statement, compare the previous Picture's height to the current Picture's height.
(D) In the ( ... ) of an `if` statement, compare the maximum height found so far to the current Picture's height.

How do you (or software like Photoshop or Gimp) make a blurry version of a digital picture?

(A) For each Pixel one at a time, find the average of that Pixel's red, green and blue intensities. That average is one number; let us assign it to the variable named `greyNum`. Then, in the new version, make each of the red, green and blue intensities of the corresponding one Pixel be set to that common value in `greyNum`.
(B) For each Pixel and for the color red, find the sum of the red intensities of that Pixel and the red intensities of certain neighboring Pixels. Then in the new version, use that sum for the red intensity of the corresponding Pixel. Repeat this computation for all the colors (green and blue as well as red) and for all the Pixels.
(C) Similar to B but also count the neighboring Pixels and divide each of the red, green and blue sums by the count to get those three intensities for the corresponding Pixel.
(D) Similar to C but multiply instead of divide.
What is the last number value that the computer copies into variable k when the computer runs the code below? In other words, what does it print?

```java
int k;
k = 1;
while( k < 5 - 1 )
{
    k = k + 1;
}
System.out.println( k );
```

Tracing with an array. The boxes on the right show what the code below before the while did, plus a little of what you must figure out. Trace rest of the using the boxes underneath to carefully determine everything the computer would have in the array elements and in variable k after each computation step, one by one. Then, select the answer that shows all the array element values at the time when all the code finishes. Use the k boxes and the blank array element boxes as if time progresses from top to bottom.

```java
int arr[] = new int[5];
for(int i=0; i<5; i++)
{   arr[i] = 2*i; }
arr[0] = 1;
int k;
k = 1;
while( k < arr.length - 1 )
{
    arr[k] = arr[k] + arr[k-1];
k = k + 1;
}
```

```
(A.)  [0] [1] [2] [3] [4]
  1   2   4   6   8

(B.)  1   3   7   6   8

(C.)  1   3   7   13  8

(D.)  1   3   7   13  21

(E.)  1   3   4   13  8
```

6 of 8 pages
One lesson from Project 06 was that an extra variable for temporary use is needed in code that swaps the values of two variable using assignments. Which code below swaps the values in int s X and Y?

(A.) int temp; X = Y; temp=X; Y=temp;
(B.) int temp; X=temp; Y=X; temp=Y;
(C.) int temp; temp=X; Y=temp; X=temp;
(D.) int temp; temp=X; Y=X; X=temp;
(E.) int temp; temp=X; Y=X; Y=temp;

Hints: After a good guess, TEST IT WITH AN EXAMPLE LIKE WHAT HAPPENS AFTER SAY X = 1; and Y = 2; Also, it's OK with the Java compiler to put 4 statements on one line, even thought it's terrible style for people.

When is a constructor method called automatically? (It is special because of the rules that the previous question was about.)

(A.) Before the new operator builds the object by allocating memory space for it.
(B.) Immediately after the new operator builds the object by allocating memory space for it.
(C.) Some time after the rest of the program starts using the object.
(D.) Never automatically: Methods only run when they are called!
Program writing (35 points) Write your work on page 2 or on a blank paper if you mess up page 2.

Write a complete Java application that does the following in the listed order:

1. (0.5 pt.) Name your app anything you want.

2. (0.5 pt.) Make a length 10 array and an array reference variable to refer to it.
   (Cheat sheet: \texttt{int[]} \texttt{A} = \texttt{new int[ 10 ];})

3. (5 pt.) (Simply input 10 \texttt{ints} into the array, thanking the user after each.) Code a loop
   (4 of 6 pt.) with loop variable \texttt{i}, where \texttt{i} will range over 0, 1, 2, ..., 9 in that order, and the
   following is done during each repetition of the body:
   a) (4 pt.) Input one integer and store it in the element of array \texttt{A} whose index is the value of \texttt{i}.
      (1) "=" means copy or store, not equals. (2) \texttt{A[ some expression ]} refers to one
      array element variable. (3) \texttt{sc.nextInt();} inputs and returns one integer value.
   b) (1 pt.) Print \texttt{Thanks} on a separate line.

4. (8 pt.) Run another (non-nested) loop that prints the 10 numbers from \texttt{A} in the reverse (4 pt.)
   order from how they were inputted. (One or multi-line printing is your choice.)
   Example: If you inputted 1, 6, 8, 4, 8, 7 etc. this loop would print etc., 7, 8, 4, 8, 6, 1

5. Demonstrate programming with one or more NON-NESTED loops to compute the values
   needed for the printing requirements a).d) below. Then make your program print, on separate
   lines, in the \texttt{exact abcd order below}: FINISH OUR printing code. (0 credit for printing unless
   the value is right; -1 ea. for forgetting to print on separate lines or getting the order wrong.)
   (It is your choice whether to incorporate the code for this (part 5.) into your loop
   for part 4., or to code the loop or loops for this (part 5.) separately.)
   a) (5 pts.) The sum of the ten \texttt{ints} in array \texttt{A}.
   b) (5 pts.) The largest among the ten \texttt{ints} in array \texttt{A} (-2pt if you find the smallest instead.)
   c) (5 pts.) Any one of the locations (that is, subscript values 0, 1, ..., 9) where the array
      element contains the largest value. That is, print the index or subscript, NOT the number
      the user had typed. We said "any of the locations" because the user might have input the
      largest \texttt{int} more than once.
   d) (5 pts.) HOW MANY TIMES the largest appears among the ten \texttt{ints} stored in \texttt{A}.

More cheat sheet tips:
\begin{itemize}
  \item \texttt{if ( TEST ) \{ \ldots (body). \} and}
  \item \texttt{if ( TEST ) \{ \ldots (body)\ldots \} else \{ \ldots (body)\ldots \}}
  \item \texttt{while( TEST ) \{ \ldots (body). \}}
  \item \texttt{for( INIT ; TEST ; UPDATE ) \{ \ldots (body). \}}
\end{itemize}

(Remember to use loop variables! and code INIT and UPDATE when you use whiles!) and

Examples of TESTs: (A "TEST" is a boolean expression.)
\begin{itemize}
  \item (1) \texttt{i < 10} \hspace{1cm} (2) \texttt{A[ i ] > biggie} \hspace{1cm} (3) \texttt{A[i] > A[j]}
\end{itemize}

How to decide which ones to use, and how to use them, in a programming questions,
cannot be summarized in a cheat sheet! That's for you to learn by practice and study.