This exam is open book. Each question is worth 3 points.

1. Are you in CS101 or CS101E?

2. Suppose \( n, j \) and \( k \) are previously defined and initialized integer variables. Write an expression that multiplies \( n \) times the quantity \( j \) plus \( k \).

3. Suppose \( n \) is a previously defined and initialized integer variable. Write an expression whose value is the remainder of \( n \) when divided by 6.

4. Write a definition for a local integer variable \( n \) whose initial value is 21.

5. Suppose \( n \) and \( m \) are previously defined and initialized integer variables. Write a statement that updates the value of \( n \) to the value of \( m \).
6. Suppose \( n \) is a previously defined and initialized integer variable. Write a statement that increases the value of variable \( n \) by 1, where the statement does not use the = operator.

7. Write a statement that defines an integer constant named \texttt{ZERO} whose value is 0.

8. Suppose \( n \) is a previously defined and initialized integer variable and that \( x \) is a previously defined and initialized floating point variable. Write a statement that updates the value of \( n \) by assigning it the int-casted value of floating point value \( x \).

9. Suppose \( j \) and \( k \) are previously defined and initialized integer variables. Write an if statement that sets variables \( j \) and \( k \) to 1 when \( j \) is less than \( k \).

10. Suppose \( j \) and \( k \) are previously defined and initialized integer variables. Write an if statement that sets variables \( j \) and \( k \) to 1 when \( j \) is less than \( k \); otherwise it sets \( j \) and \( k \) to 10.
11. Suppose \( j \) and \( k \) are previously defined and initialized integer variables. Write an if-else-if statement that sets variable \( j \) to 1, 2, or 3 respectively depending whether \( j \) is less than \( k \), \( j \) equals \( k \), or otherwise (i.e., \( j \) is greater than \( k \)).

12. Write a for statement that iterates 1000 times using a previously undefined integer index variable \( i \). Each iteration of the loop should display the current value of \( i \) (i.e., the values to be displayed are 0 …999).

13. Define a Scanner variable \( stdin \) associated with the standard input stream. Your definition should be compatible with the latest version of Java.

14. Define a Scanner variable \( stream \) associated the string represented by the previously defined and initialized String variable \( s \). Your definition should be compatible with the latest version of Java.
15. Suppose $s$ and $t$ are previously defined and initialized $\texttt{String}$ variables. Write a code segment that causes $t$ to represent the substring of $s$ defined by indices 3 through 7 inclusive.

16. Suppose $\texttt{input}$ is a previously defined and initialized $\texttt{Scanner}$ variable. Write a $\texttt{while}$ loop that iterates while the input stream associated with variable $\texttt{input}$ has unread values. Each iteration of the loop should read and display the next $\texttt{int}$ value.

17. In TWENTY words or less how do the $\texttt{==}$ operator and the $\texttt{\texttt{equals}()}$ method typically differ.

18. Define a new $\texttt{int}[]$ variable $\texttt{data}$ that references a list composed of the values 1, 2, and 3.

19. Define a new $\texttt{int}[]$ variable $\texttt{data}$ that references a new array with ten elements.

20. Update an initialized $\texttt{int}[]$ variable $\texttt{data}$ so that it references a new array with zero elements.
21. Suppose list is a defined and initialized int[] variable with 100 elements. Write a statement that assigns 10 to the first element of list.

22. Suppose list is a defined and initialized int[] variable with 100 elements. Write a statement that assigns 10 to the last element of list.

23. Suppose list is a defined and initialized int[] variable. Write a statement that displays the number of elements represented by list.

24. Suppose list is a defined and initialized String[] variable for an array with n elements and result is a defined and initialized String variable. Write a code segment that causes result to reference a String whose value is a concatenation of the values represented by list.

25. What is the output of the following program?
   
   public class Think {
       public void static mystery(int n) {
           n = 5;
       }

       public void static main(String[] args) {
           int n = 0;
           mystery(n);
           System.out.println("n = " + n);
       }
   }

   The output is
   
   n = __________
The remaining questions deal with a class named \texttt{Car}. Class \texttt{Car} is to have the following private \texttt{double}-valued attribute.

- \texttt{speed} – current travelling rate of the car in miles per hour.

Class \texttt{Car} is to also have the following \texttt{public} methods.

- \texttt{Car()} – default constructor that configures a car to have a speed of 55 miles per hour.
- \texttt{Car(double v)} – specific constructor that has a single \texttt{double}-valued parameter \(v\). The constructor configures the new car to have a speed of \(v\).
- \texttt{howFar(double t)} – indicates how far the car will go if it travels for \(t\) hours, where \(t\) is its \texttt{double}-valued parameter; i.e., the return value is the product of \(t\) and the speed attribute of the car.
- \texttt{getSpeed()} – returns the current speed of the car.
- \texttt{setSpeed(double v)} – sets the current speed of the car to the value of its single \texttt{double}-valued parameter \(v\).
- \texttt{clone()} – returns a new \texttt{Car} with the same attribute of this car.
- \texttt{equals(Object)} – returns whether its \texttt{Object}-valued parameter value is a \texttt{Car} with the same speed as this car.
- \texttt{toString()} – returns a \texttt{String} representation of the car. The representation should consist of the value of the speed attribute within parentheses.

26. Implement the \texttt{Car} default constructor using an assignment statement to configure the speed attribute.

27. Implement the \texttt{Car} speed accessor \texttt{getSpeed()}.

28. Implement the \texttt{Car} speed mutator \texttt{setSpeed()}. 
29. Implement the `Car` specific constructor using a mutator to configure the speed attribute.

30. Implement the `toString()` method for `Car`.

31. Implement the `Car` default constructor so that it uses the specific constructor to configure the speed attribute of the new `Car`.

32. Implement the `equals()` method for `Car`.
33. Implement the `clone()` method for `Car`.

34. Suppose `list` is a `Car[]` variable. Write a statement that updates the `list` element with index `i` so that its speed is now 25 miles per hour.

Pledge: