This exam is open text book and closed notes. Different questions have different points associated with them. Because your goal is to maximize your number of points, we recommend that you do not spend too long on any particular question during your first pass through the exam.

Assume stdin is an initialized Scanner throughout the test.

In questions asking whether two object references are the same we are asking whether they represent like objects (i.e., we are not asking whether the references point to the same memory location).

1. (5 points) What section of CS101 are you in?

   ______  2 CS101E
   ______  3 0800-0915 Thursday
   ______  4 0930-1045 Thursday
   ______  5 1100-1215 Thursday
   ______  6 1230-1345 Thursday
   ______  7 1400-1515 Thursday
   ______  8 1530-1645 Thursday
   ______  9 1700-1815 Thursday
   ______ 10 1830-1945 Thursday
   ______ 11 2000-2115 Thursday

Pledge:
2. (6 points) What are the values of the following expressions?

\[
\begin{align*}
\text{true} & \land \text{false} \\
\text{true} & \lor \text{false} \\
\neg \text{true} \\
\neg \text{true} & \land \text{false} \\
(3/4) \neq 0 \\
1 + 2 & < 1 + 4
\end{align*}
\]

3. (3 points) Suppose method \texttt{evaluate()} is a method that takes two \texttt{boolean} parameters and returns a \texttt{boolean} value according to the following truth table.

<table>
<thead>
<tr>
<th>\text{p}</th>
<th>\text{q}</th>
<th>\text{result}</th>
</tr>
</thead>
<tbody>
<tr>
<td>false</td>
<td>false</td>
<td>true</td>
</tr>
<tr>
<td>false</td>
<td>true</td>
<td>false</td>
</tr>
<tr>
<td>true</td>
<td>false</td>
<td>true</td>
</tr>
<tr>
<td>true</td>
<td>true</td>
<td>false</td>
</tr>
</tbody>
</table>

What are the values of the following invocations?

\[
\begin{align*}
\text{evaluate(true, false)} \\
\text{evaluate(false, true)} \\
\text{evaluate(false, false)}
\end{align*}
\]

4. (4 points) Define and initialize a \texttt{boolean} Java variable \texttt{isSilent} that will be used to maintain the ring mode of a cell phone. The cell phone in question is currently not allowed to ring.

5. (5 points) Use an appropriate \texttt{if} statement to complete the following code segment that displays \texttt{wahoo} only if the extracted integer is \texttt{not equal} to 88.

```java
System.out.print("Enter a number: ");
int n = stdin.nextInt();
```

```java
________________________________________________________________
________________________________________________________________
________________________________________________________________
```
6. (8 points) Complete the following code segment that displays *wahoo* if the extracted integer is zero and displays *herring* otherwise. (We believe your answer should not need all of the provide space.)

```
System.out.print("Enter a number: ");
int n = stdin.nextInt();

________________________________________________________________
________________________________________________________________
________________________________________________________________
________________________________________________________________
________________________________________________________________
________________________________________________________________
________________________________________________________________
________________________________________________________________
```

7. (4 points) Consider the following method `mystery`().

```
public static boolean mystery(boolean p, boolean q) {
    if ( p == q ) {
        return true;
    }
    else if ( q ) {
        return true;
    }
    else {
        return false;
    }
}
```

What truth table is implemented by method `mystery`()?

<table>
<thead>
<tr>
<th></th>
<th></th>
<th>result</th>
</tr>
</thead>
<tbody>
<tr>
<td>false</td>
<td>false</td>
<td>________</td>
</tr>
<tr>
<td>false</td>
<td>true</td>
<td>________</td>
</tr>
<tr>
<td>true</td>
<td>false</td>
<td>________</td>
</tr>
<tr>
<td>true</td>
<td>true</td>
<td>________</td>
</tr>
</tbody>
</table>
8. (5 points) Consider the following code segment.

```java
if ( i > j ) {
    if ( j > k ) {
        System.out.println("A");
    }
    else if ( k == i ) {
        System.out.println("B");
    } else {
        System.out.println("C");
    }
} else if (j > k ) {
    System.out.println("D");
} else {
    if ( i == k ) {
        System.out.println("E");
    } else {
        System.out.println("F");
    }
}
```

What is the output if \(i = 3, j = 2,\) and \(k = 1?\)

_____________

What is the output if \(i = 2, j = 4,\) and \(k = 3?\)

_____________

What is the output if \(i = 2, j = 1,\) and \(k = 3?\)

_____________

What is the output if \(i = 2, j = 1,\) and \(k = 4?\)

_____________

What is the output if \(i, j,\) and \(k\) all have same value?

_____________

9. (4 points) What is the most misleading part of the following legal code segment?

```java
System.out.print("Enter a number: ");
int n = stdin.nextInt();
int i = 0;
while (i < n )
    ++i;
    System.out.println("A");
System.out.println("B");
```

10. (5 points) The following code segment does not compile. Explain why the code segment cannot display the value of \(n\).
(Your answer should not mention infinite loops, missing input, or missing prompts.)

```java
while ( stdin.hasNextInt() ) {
    int n = stdin.nextInt();
    System.out.println(n);
}
```
11. (5 points) Consider the following code segment.

```java
System.out.print("Enter a number: ");
int m = stdin.nextInt();
System.out.print("Enter a number: ");
int n = stdin.nextInt();

int i = m;

while ( i < n ) {
    System.out.println("A");
    ++i;
}
```

How many times is A displayed if the values supplied for m and n are respectively 1 and 2?

_____________

How many times is A displayed if the values supplied for m and n are respectively 0 and 3?

_____________

How many times is A displayed if the values supplied for m and n are respectively 2 and 1?

_____________

How many times is A displayed if the values supplied for m and n are respectively –1 and 0?

_____________

How many times is A displayed if the same value is supplied for both m and n?

_____________

12. (4 points) Examine the following definition of a Tune constructor

```java
public Tune() {
    String artist = "Thomas Alva Edition";
    String title = "Mary had a little lamb";
    int year = 1877;
}
```

and explain why the following code segment

```java
Tune fave = new Tune();
System.out.println( fave );
```

produces output

```java
Tune( null, null, 0 )
```
13. (7 points) Give a definition for \texttt{Tune} method \texttt{getYear()} where the method body consists of a single statement.

________________________________________________________________
________________________________________________________________
________________________________________________________________

14. (7 points) Complete the following definition for a \texttt{Tune} constructor, which configures the new object based on its parameters. The constructor body must not have any variable definitions or assignment statements.

\begin{verbatim}
public Tune(String performer, String name, int year) {

________________________________________________________________
________________________________________________________________
________________________________________________________________

}
\end{verbatim}

15. (7 points) Complete the definition for \texttt{Tune} method \texttt{sameArtist()} with a single \texttt{Tune} parameter named \texttt{that}. The method returns true only if the current object being accessed (the \texttt{this} object) and \texttt{that} have the same performer (See page 1 for a definition of \texttt{same}).

\begin{verbatim}
public boolean sameArtist(Tune that) {

String thisPerformer = ________________________________

String thatPerformer = ________________________________

if ( ________________________________ )
    return true;
}
else {
    return false;
}
\end{verbatim}
16. (7 points) Use an appropriate if-else statement to complete the definition for \texttt{Tune} method \texttt{setYear()} with its single \texttt{int} parameter \texttt{y}. If \texttt{y} is at least 1, then \texttt{y} is the new value for the \texttt{year} attribute of the \texttt{Tune}. Otherwise, the \texttt{year} attribute is unchanged and a message is displayed rejecting the update.

\begin{verbatim}
public void setYear(int y) {
    if (y >= 1) {
        // Update year attribute
        year = y;
    } else {
        // Display message rejecting update
        System.out.println("Year update rejected.");
    }
}
\end{verbatim}

17. (2 points) TRUE FALSE The default constructor for class \texttt{Tune} takes three parameters.

18. (2 points) TRUE FALSE \texttt{Tune} method \texttt{setTitle()} is a mutator instance method.

19. (2 points) TRUE FALSE Variable \texttt{title} is an instance variable in method \texttt{setTitle()}.

20. (2 points) TRUE FALSE Variable \texttt{performer} is an instance variable in method \texttt{toString()}.

21. (2 points) TRUE FALSE Variable \texttt{track} is a formal parameter in method \texttt{setTitle()}.

22. (2 points) TRUE FALSE The below code compiles as the method \texttt{main()} in a program \texttt{Access.java}.

\begin{verbatim}
public static void main(String[] args) {
    Tune tune = new Tune();
    String s = tune.title;
}
\end{verbatim}

23. (2 points) TRUE FALSE The below code compiles as the method \texttt{main()} in a program \texttt{Access.java}.

\begin{verbatim}
public static void main(String[] args) {
    Tune tune = new Tune();
    String s = tune.getTitle();
}
\end{verbatim}