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).

Page 1	_____ /	5
Page 3	_____ /	18
Page 4	_____ /	12
Page 5	_____ /	14
Page 6	_____ /	9
Page 7	_____ /	21
Page 8	_____ /	21
Page 8	_____ /	20
Total	_____ /	100

Pledge:

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

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

```
true && false          _____
true || false          _____
! true                 _____
! true && false        _____
( 3 / 4 ) != 0         _____
1 + 2 < 1 + 4          _____
```

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

p	***q***	***result***
false	false	true
false	true	false
true	false	true
true	true	false

What are the values of the following invocations?

```
evaluate(true, false)     _____
evaluate(false, true)     _____
evaluate(false, false)    _____
```

4. (4 points) Define and initialize a `boolean` Java variable `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 **`if`** statement to complete the following code segment that displays *wahoo* only if the extracted integer is *not equal* to 88.

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

_____

_____

_____
```

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()`?

p	*q*	*result*
false	false	_____
false	true	_____
true	false	_____
true	true	_____

8. (5 points) Consider the following code segment.

```
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?

```
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.)

```
while ( stdin.hasNextInt() ) {
   int n = stdin.nextInt();
}
System.out.println(n);
```

11. (5 points) Consider the following code segment.

```
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

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

and explain why the following code segment

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

produces output

```
Tune( null, null, 0 )
```

13. (7 points) Give a definition for `Tune` method `getYear()` where the method body consists of a single statement.

_____

_____

_____

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

```
public Tune(String performer, String name, int year) {
```

_____

_____

_____

```
}
```

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

```
public boolean sameArtist(Tune that) {

    String thisPerformer = _____

    String thatPerformer = _____

    if ( _____ )
        return true;
    }
    else {
        return false;
    }
}
```

16. (7 points) Use an appropriate if-else statement to complete the definition for `Tune` method `setYear()` with its single `int` parameter `y`. If `y` is at least 1, then `y` is the new value for the `year` attribute of the `Tune`. Otherwise, the `year` attribute is unchanged and a message is displayed rejecting the update.

```
public void setYear(int y) {

    _____

    _____

    _____

    _____

    _____

    _____

}
```

17. (2 points) TRUE FALSE The default constructor for class `Tune` takes three parameters.

18. (2 points) TRUE FALSE `Tune` method `setTitle()` is a mutator instance method.

19. (2 points) TRUE FALSE Variable `title` is an instance variable in method `setTitle()`.

20. (2 points) TRUE FALSE Variable `performer` is an instance variable in method `toString()`.

21. (2 points) TRUE FALSE Variable `track` is a formal parameter in method `setTitle()`.

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

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

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

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