Name____________________________

Q1(**4 pts**): Consider the array:

```
s[ 0 ] = 7
s[ 1 ] = 0
s[ 2 ] = -12
s[ 3 ] = 9
s[ 4 ] = 10
s[ 5 ] = 3
s[ 6 ] = 6
```

(a). The value of *s.length* is:

(b). The value of `s[ s[ 6 ] - s[ 5 ] ]` is:

(c). The value of `s[ s[ 2 ] ] + 12 is:`

(d). Write a Java code segment (using *for* loop statement) to calculate the sum of all array elements.

Q2(**3 pts**): Consider the program below:

```
public class Test
{
   public static void main( String args[] )
   {
      int a[];
      a = new int[ 5 ];

      for ( int i = 0; i < a.length; i++ )
         a[ i ] = i;


      int result = 0;
      for ( int i = 0; i < a.length - 1; i++ )
         a[i] = a[ i + 1];


      System.out.printf( "%d\t\%d\n", a[0], a[a.length - 1]);
   } // end main
} // end class Test
```

The output of this program will be:

Q3(**1.5 pts**): Write a Java statement that assign random integers to the variable ***n*** in the range **0** <= **n** <= **10**.

Q4(**2.5 pts**): What is output by the following Java code segment?

```
int temp;
temp = 50;

if ( temp > 70 )
{
   System.out.println( "Too hot." );

   temp = temp – ( temp > 150 ? 100 : 20 );
}
else
{
   if ( temp < 60 )
   {
      System.out.print("Too cold.");

      temp = temp + (temp < 50 ? 30 : 20);
   }
}
if ( temp >= 65 )
   System.out.println( "Not cold!" );
else
   System.out.println( “Not hot!” );
```

Q5(**1 pts**): How many times is the body of the loop below executed?

```
int counter;
counter = 20;

while ( counter >= 20 )
{
   counter = counter + 1;

} // end while
```

Q6(**3 pts**): Consider the following two Java code segments. What will be the values for ***i*** and ***j*** after each of two Java code segments have been executed respectively?

Segment 1

```
int i = 0;


while ( i < 15 )
{
   i++;
   System.out.println( i );
}
```

Segment 2

```
int j = 0;

for ( j = 0; j <= 15; j++ )
{
   System.out.println( j );
}
```

Q7(**2 pts**): What will be the output when the following code segment has been executed?

```
int q = 5;
switch( q )
{
   case 1:
     System.out.println( 1 );
   case 2:
     System.out.println( 2 );
   case 3:
     System.out.println( 3 );
   case 4:
     System.out.println( 4 );
   case 5:
     System.out.println( 5 );
   default:
     System.out.println( "default" );
} // end switch
```

Q8(**4 pts**): What does the following program print?

```
 public class Mystery3
{
          public static void main( String args[] )

          {
                    int row;
                    int column;

                    for ( row = 3; row >= 1; row-- )
                    {
                              column = 1;
                              while ( column <= 4 )
                              {
                                        System.out.print( row % 2  ==  0  ?  "+"  :  "*" );
                                        ++column;
                              }
                              System.out.println();
                    }
           }
 }
```

Q9(**1 pts**): Declare and create an array of 10 elements of type double.

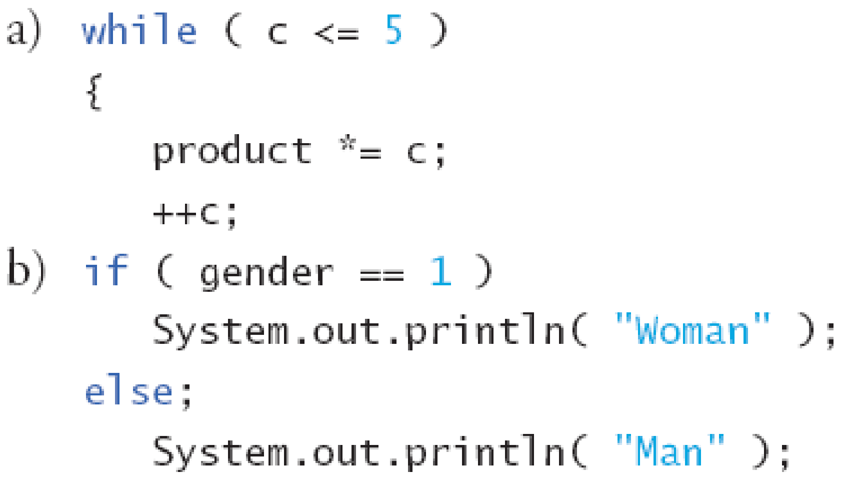
Q10(**1.5 pts**): Find and correct error(s) in the following code segment.

Assume int b[] = new int[10];

```
for ( int i = 0;  i <= b.length(),  i++)
        b[ i ] = i;
```

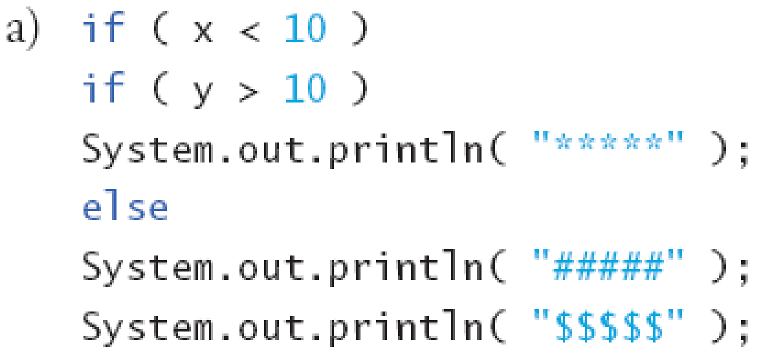
Q11(**1.5 pts**): Write a Java statement that assign random integers to the variable ***n*** in the range **-1** <= **n** <= **1**.

Q12(**2 pts**): Identify and correct the errors in each of the following sets of code:



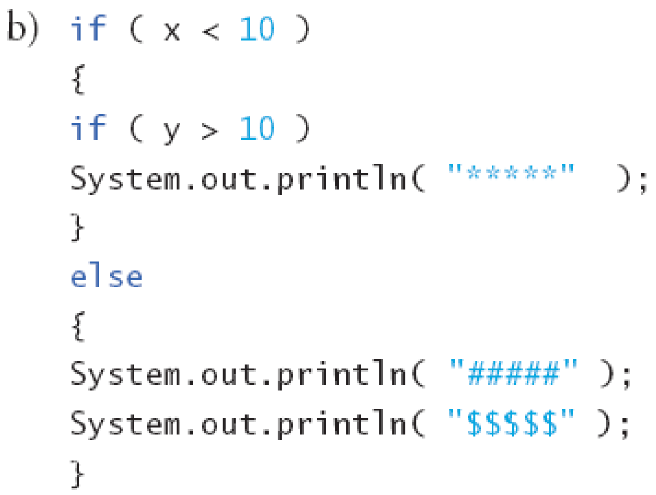
```
a) while ( c <= 5 )
   {
      product *= c;
      ++c;
b) if ( gender == 1 )
      System.out.println( "Woman" );
   else;
      System.out.println( "Man" );
```

Q13(**4 pts**): Given the following sets of code:



a)

```
if ( x < 10 )
if ( y > 10 )
System.out.println( "*****" );
else
System.out.println( "#####" );
System.out.println( "$$$$$" );
```



b)

```
if ( x < 10 )
{
if ( y > 10 )
System.out.println( "*****"  );
}
else
{
System.out.println( "#####" );
System.out.println( "$$$$$" );
}
```

(1) Determine the outputs when **x = 9 and y = 11**

a) **b)**

(2) Determine the outputs when **x = 11 and y = 9**.

a) **b)**

Q14(**2 pts**): Write a Java code segment (using ***for***… statement) that sums the odd integers from 7 to 999 (including 7 and 999).

Q15 (**2 pts**): What is output by the following code segment?

```
for ( int i = 0; i <= 11; i++ )
{
    if ( i % 2 == 0 )
    continue;

    if ( i == 11 )
        break;

    System.out.printf( "%d ", i );
} // end for
```

Q16(17 pts): For each of the following problems, write a Java program segment that performs the specified action.

1. Write a `for` loop that prints all the even integers from 2 to 1002, inclusive.

2. Write a `do…while` loop that prints the integers from 10 to 0, inclusive.

3. Write a `for` loop that counts from 1 to 5. Use a `switch` statement to display a letter in the alphabet that corresponds to the number (i.e., 1 is A, 2 is B, etc.).

4. Write a `while` loop that sums the integers from 1 to 10, excluding 3 and 6. Print the sum.

5. Write a `for` loop that attempts to display the numbers from 1 to 11, but terminates when the control variable reaches the value 6.

6. Write a `for` loop to display the numbers from `1` to `12`, but skip the value `7` by using a `continue` statement.

7. Write a method that takes an integer as an argument and returns the remainder of that value divided by 5.

8. Write a method `multiple` that takes two integers as its arguments and returns `true` if the first integer is divisible evenly by the second one (i.e., there is no remainder after division); otherwise, the method should return `false`.

9. Write a method `halve` that takes one floating-point value of type `double` as its argument and returns the value of that number divided by 2.

Q17(13 pts): Coding Exercises (xxCh8 Class Square).

1. Write the class declaration for class `Square` that has a `private` instance variable `side` of type `double` and a no-argument constructor that sets the `side` to `1.0` by calling a method named `setSide` that you will declarein *Coding Exercise 2*.

2. Write a method `setSide` for the class you defined in *Coding Exercise 1*. Set the `side` variable to the argument of the method. Also make sure that the `side` is not less than `0.0`. If it is, keep the default setting of `1.0`.

3. Write a method `getSide` for the class you modified in *Coding Exercise 2* that retrieves the value of instance variable `side`.

4. Define another constructor for the class that you modified in *Coding Exercise 3* that takes one argument, the `side`, and uses the `Square`'s *set* method to set the `side`.

5. Write a method `computeArea` for the class that you modified in *Coding Exercise 4* that computes the area of a `Square`.

6. Define a `toString` method for the class that you modified in *Coding Exercise 5* that will return a `String` containing the value of `side` and the area of the `Square`.

7. Draw a UML class diagram for the class Square from the above exercises.