What is your name?: Key (0 points)

There are three sections:
   I. True/False ...............  .30 points; (15 questions, 2 points each)
   II. Multiple Choice ..........  .30 points; (5 questions, 6 points each)
   III. Short Answer ..........  .40 points; (5 questions, 8 points each)

100 points total

This test is worth 15% of your final grade. For the true/false and multiple choice problems you must put your answers on the bubble form. Put your answers to the short answer questions on these sheets, which must also be turned in. This test is open book and open notes. You have 50 minutes.

• For the True/False questions, if the answer is True, fill in a on your answer form. If the answer is False, fill in b on your answer form.
• For the multiple choice problems, select the best answer for each one and select the appropriate letter on your answer sheet.
• Be careful - more than one answer may seem to be correct. Some questions are tricky.

I. True/False: (2 points each)

T F 1. When implementing a menu in your code, good programming practice suggests that a while loop is preferable to a do loop.

T F 2. When implementing an infinite loop in your code, good programming practice suggests that a while loop is preferable to a do loop.

T F 3. The output of the code below is the value: 49

```java
int x=2, y=5, z=7;
System.out.println(x + y * z); // 2+(5*7) = 37
```

T F 4. Assume the code shown below, where method swapValues is called. Output of this segment of code is: Values are: 3 5

```java
int x = 3;
int y = 5;
swapValues(x, y);
System.out.println("Values are: "+x+" "+y);
// ... other code

public void swapValues(int num1, int num2)
{
    int temp = num1;
    num1 = num2;
    num2 = temp;
}
```

Missed this
5. The following code prints the words: First

```java
String word = "First";
switch (word) {
    case ("First"): System.out.println("First one"); break;
    case ("Second"): System.out.println("Second one"); break;
    case ("Third"): System.out.println("Third one"); break;
}
```

6. You can have overloaded methods that have the same name but different argument signatures, however you can only have a single constructor for a class.

7. Having a return type with a Constructor allows the user to chain together the creation of multiple objects, where an existing object is used to initialize a new object.

8. The name of a Java class (e.g. Date) should be the same as the filename (e.g. Date.java)

9. If you don’t implement the toString() method in a class, then using an instance of that class in a System.out.println(...) statement gives a compiler error.

10. The output of the program segment below is the text: Nope ***

```java
boolean x = false;
if ( x = true)
    System.out.print("Yup");
else
    System.out.print("Nope");
    System.out.print(" *** ");
```

11. In Java it is possible to write a method with a return type of void that receives a value as a parameter, changes that parameter value, and has that value reflected back to the calling part of the program.

12. Assuming we have a String named theWord, in Java we can check for its size using: `theArray.size should be theArray.length`

13. Every method must have the return type declared unless the method doesn’t return anything, in which case the return type can be left off.

14. A static method can not access non-static instance variables unless the (nonStatic) declaration is used in front of the instance variables.

15. The following code will give the output: immediately

```java
String words1 = new String("eliminate all bigots immediately");
mid = word1.substring(word1.indexOf(' ') +1 , word1.length());
last = mid.substring(mid.indexOf(' '), mid.length());
last = last.substring(last.indexOf(' '), last.length());
System.out.println(last);
```

Output: bigots immediately
16. Assume you are doing a numbers guessing game, where you are attempting to guess a number between 0 and 3000 (this is a binary search). After each guess, you are told if you need to guess higher or lower. What is the maximum number of guesses you would need to find the number?

(a) 10
(b) 11
(c) 12
(d) 13
(e) None of the above

\[ 1 2 3 4 5 6 7 8 \leq 2^3 \text{ values} \]
\[ \uparrow \uparrow \uparrow \uparrow \]
3 + 1 = 4 guesses max
\[ 2^n = 2048 \leq 3000 \]
\[ 2^{12} = 4096 \]

17. Consider using a bubble sort (as discussed in class) to sort 8 numbers. What is the maximum number of pairs of values that need to be swapped before the numbers are sorted?

(a) 7
(b) 28
(c) 35
(d) 56
(e) None of the above

\[ \underline{8} \underline{6} \underline{5} \underline{4} 3 2 1 \]
\[ + 6 \]
\[ + 5 \]
\[ + 4 \]
\[ + 3 \]
\[ + 2 \]
\[ + 1 \]
\[ \underline{28} \]

18. Consider the program method given below.

```java
public void problem18( int[] theNumbers) {
    int x = theNumbers[0];
    for (int i=0; i<theNumbers.length; i++) {
        if (theNumbers[i] > x)
            x = theNumbers[i];
    }
    System.out.println( x);
}
```

Its output is:

(a) The minimum value in array theNumbers
(b) The maximum value in array theNumbers
(c) The average value in array theNumbers
(d) The sum of the values in array theNumbers
(e) None of the above
19. What is the output of the code given in the two columns below when method `startUp()` is called?

```java
public class Confuse {
    private int x; // Constructor
    private int y;

    public Confuse() {
        x = 3; y = 2;
    }

    private void first(int s, int t) {
        x += 4; y += 3;
    }

    private void second(int y) {
        x += 2; y = 3;
    }

    private void third(int s, int t) {
        setXY(s, t);
        y = 3; t = 4;
    }

    private void setXY(int x, int y) {
        this.x = x;
        this.y = y;
        second(y);
    }

    private void display() {
        System.out.println(x + " * " + y);
    }

    public void startUp() {
        int x=0, t=0;
        first(x, y);
        second(x);
        third(x, y);
        display();
    }
}
```

Output: 2 5

a) 2 4
b) 4 5
c) 8 9
d) 5 6
e) None of the above

20. What is the output of the code given below when `problem20Driver()` is called?

```java
class Confuse {
    private int x; // Constructor
    private int y;

    public Confuse() {
        x = 3; y = 2;
    }

    private void first(int s, int t) {
        x += 4; y += 3;
    }

    private void second(int y) {
        x += 2; y = 3;
    }

    private void third(int s, int t) {
        setXY(s, t);
        y = 3; t = 4;
    }

    private void setXY(int x, int y) {
        this.x = x;
        this.y = y;
        second(y);
    }

    private void display() {
        System.out.println(x + " * " + y);
    }

    public void startUp() {
        int x=0, t=0;
        first(x, y);
        second(x);
        third(x, y);
        display();
    }
}
```

```java
public void problem20Driver() {
    String phrase = "Eschew surplusage";
    // create character array from String
    char[] theArray = phrase.toCharArray();

    problem20( theArray );

    // display array contents
    System.out.print( theArray );
}
```

```java
public void problem20(char[] w) {
    int x = w.length;
    char c;
    for (int i=0; i<x; i++) {
        c = w[i];
        w[i] = w[x-i-1];
        w[x-i-1] = c;
    }
}
```

a) The contents of the original array in reverse order
b) The contents of the original array in the original order
c) The contents of the original array with half of the characters reversed
d) The original array with characters rearranged so they are neither in the original nor reversed order
e) None of the above
I. Short Answer (8 pts. each)

Some problems ask: "What does a program do?" This does not mean giving a description of each line, but rather when taken overall, describing what is the effect of the code.

21. What does the following code do when methodA is called?

```java
public int methodB( int x, int n) {
    if (n == 0)
        return 0;
    else
        return x + methodB( x, n-1);
}
public int methodA( int x) {
    return methodB( x, x);  
}
```

22. What does the following code do?

```java
double methodC( double x, int n) {
    if (n == 0)
        return 1.0;
    else if (n > 0)
        return methodC( x, n - 1) * x;
    else {
        System.out.println("Error!");
        return 0.0;
    }
}
```

23. Consider the Date class we discussed and developed in class. Imagine you are writing this class yourself. When displaying an instance of Date using a System.out.println statement, assume just a number appears instead of the month, day, and year information you would like. What is likely the problem?

the `toString()` method has not been implemented.
24. Consider the *Date* class (that we discussed in class). Assume that there is also an *Employee* class, where each employee has an instance variable called *startDate* that is of type *Date*. Assume the code below compiles and runs:

```java
class EmployeeDriver {
    // ... the following code would be in some method ...
    Employee e1 = new Employee( "Jack", new Date(11,7,2006));
    Employee e2 = new Employee( e1);
    e1.setName( "Jill");
    // Now display both employees
    System.out.println("Employees are: " + e1 + "\n" + e2);
    // ...
}
```

The output of this code displays two employees (Jack and Jill), however the dates for both of them are the same. Why is this?

The copy constructor used to create e2 does a shallow copy of Jack's date, so the date is shared. Changing it for Jack also changes it for Jill.

25. Consider the two classes given below:

```java
class ClassE {
    private int a;
    public ClassE(int value) {
        a = value;
        if (value < 4) {
            ClassD instance1 = new ClassD( value);
        }
        System.out.print(" " + a);
    }
    public void method1(int a) {
        a = a + 3;
    }
}
```

What is the output of the code shown above when creating an instance of ClassD using:

```java
    ClassD classDInstance = new ClassD( 2);
```

Please circle your answer.