1. (10 points) What are five Java statements we have covered so far? message, return, assignment, for, while, if, switch, block, empty

2. (5 points) What is Java for the last char in a String, s? s.charAt(s.length()-1)

3. (5 points) What is the syntax of an assignment statement? Use BNF or reasonable facsimile thereof.
   \[
   \text{<assignment statement>} ::=: \text{<variable>} = \text{<expression>}
   \]

4. (5 points) Assume you have a variable, classDaysLeft, which has in it the number of days left in the semester. Write Java code to `sout “School’s out!”` if the value of classDaysLeft is zero, otherwise `sout` the number of days remaining along with some informative text (like, “Days left=”).
   ```java
   if(classDaysLeft<=0)
     sout("School's out!");
   else sout("days left=" + classDaysLeft);
   ```

5. (5 points) Write a method, named year, which is passed an `int` parameter telling how many credits a student has completed and `souts` "senior" if they have at least 23 credits, "junior", if they have 15-22 credits, "sophomore" if 7-14, and "first year" if less than 7.
   ```java
   void year (int cred){
     if (cred>=23) sout("senior");
     else if (cred>=15)sout("junior");
     else if (cred>=7)sout("sophomore");
     else sout("first year");
   }
   ```

6. (10 points) Write code to `sout` the numbers from 1-10000, one per line.
   ```java
   for (int i=0; i<10000; sout(""+i++)); // empty body??
   ```

7. (5 points) What is the return type of `emit` (see #9)? void

8. (5 points) What is a method with no return type called? a constructor

9. (5 points) Here is the `emit` method from the formatter example from class. In the interest of political correctness (or politeness, or something...) modify it so that it censors all four-letter words. Mark it up so that 4-letter words are simply thrown away.
   ```java
   static void emit(String nextChunk) {
     if (nextChunk.length()==4) return; // 4-letter words just disappear!
     if (!fits(nextChunk)) {
       flush();
     }

     buffer += nextChunk;
   }
   ```

10. (5 points) Here is some broken code:
    ```java
    class Exam {
      int x;

      void setX() { // spurious parameter deleted...
        x = 17;
      }

      void printX() {
        System.out.println("x=" + x);
      }

      public static void main(String[] adf) {
        setX();
        printX();
      }
    }
    ```

    When it is executed, to the programmer’s horror, it prints \(x=0\). Why? Shadowing!
11. (5 points) Write a method, named abs, which returns the absolute value of its int parameter (e.g. abs(-7)=abs(7)=7).

```java
int abs(int x) {
    if (x>0)
        return x;
    return -x;
}
```

12. (5 points) Write a method, isVowel, that returns true or false on the basis of whether its char parameter is a vowel.

```java
boolean isVowel(char ch) {
    return ch=='a' || ch=='e' || ch=='i' || ch=='o' || ch=='u';
}
```

13. (10 points) Write a method, deleteVowels, that returns its String parameter with the vowels deleted. If the parameter were "heffalump", it should return "hfflmp". Assume you have a working isVowel method (above).

```java
String deleteVowels(String s) {
    String returnMe = "";
    for (int i=0; i<s.length(); i++)
        if (!isVowel(s.charAt(i))
            returnMe += s.charAt(i);
    return returnMe;
}
```

14. (5 points) Add code so that this method will `sout` how many times the countButton has been pressed, (each time you press it).

```java
int count;
private void countButtonActionPerformed(java.awt.event.ActionEvent evt) {
    count++;
    sout("button pushed " + count + " times.");
}
```

15. (5 points) When is an infinite loop a good thing? I.e. when would you write one on purpose? When you are writing stopwatch or simulation/animation code.

16. (10 points) Write a complete Account class with an int balance variable and its accessors.

```java
class Account {
    int balance;
    void setBalance(int b) {balance = b;}
    int getBalance() {return balance;}
}
```

17. (10 points extra credit) Assume that Account has a balance variable with an accessor. Write a method, richest, which is passed an ArrayList<Account> and returns the Account with the biggest balance.

```java
Account richest (ArrayList<Account> list) {
    Account max=list.get(0);
    for (Account nextAccount: list) {
        if (nextAccount.getBalance() > max.getBalance())
            max = nextAccount;
    }
    return max;
}
```