1. (10 points) What are five built-in Java classes you have used? See review sheet!
   a) 
   b) 
   c) 
   d) 
   e) 

2. (5 points) Why is building a prototype such an effective problem-solving technique? It makes the problem simpler and allows you to build cognitive structure for the problem at hand.

3. (5 points) What is the syntax of an if statement? Use BNF or reasonable facsimile thereof. See text!

4. (5 points) What are the semantics of an if statement (refer to your BNF above)? See text!

5. (4 points) Why do we write toString as part of every class? I.e. what is it used for? So you can see what's inside the instances. For debugging.

6. (7 points) Write code to `sout` the odd numbers from 1-1234, one per line.
   
   ```java
   for (int i = 1; i < 1234; i += 2) sout (**+i);
   ```

7. (5 points) What methods have no return type? `void`

8. (6 points) What are the three steps of instantiation?
   i allocate storage for variables
   ii initialize variables (to 0 by default)
   iii execute constructor
9. (20 points) Write a Star class with two instance variables, name and income (in whole! dollars); accessors for income, and the standard toString method.

```java
public class Star {

    protected int income;
    protected String name;

    public Star() {  //empty default constructor
        this();     //invoke the default constructor
        this.income = income;
        this.name = name;
    }

    public int getIncome() { return income; }
    public String getName() { return name; }

    public void setIncome(int income) { this.income = income; }
    public void setName(String name) { this.name = name; }

    public String toString() {
        String returnMe = "I am a Star: ";
        returnMe += "\tincome= " + getIncome();
        returnMe += "\tname= " + getName();
        return returnMe;
    }   // toString()
}  // Star
```
10. (5 points) Write a method, named level, with a Star parameter, which returns the type of star the Star is. If their income is less than 100,000, return "minor", 100,000-1,000,000, "major", over 1 million, "super!".

String level(Star aStar)
{
    if (aStar.getIncome() < 100000)
        return "minor";
    else if (aStar.getIncome() > 1000000)
        return "super!";
    else return "major";
}

11. (6 points) Here is the emit method from the formatter example from the text. 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.

static void emit(String nextChunk) {
    if (nextChunk.length() == 4)
        return;   // skip if 4 letters
    if (!fits(nextChunk)) {
        flush();
    }

    buffer += nextChunk;
}

12. (4 points) Here is some broken code:

class Exam {
    Controller theController;

    Exam() {
        Controller theController = new Controller(this);
        sout("??");
    }

    public void paint(Graphics g) {
        theController.paint(g);
    }
}

When it is executed, to the programmer's horror, it throws a NullPointerException in paint(). To see if theController was initialized the programmer added the sout line; and it did output ??'. What's the problem?

theController in Exam() shadows the instance variable; delete Controller on line 5.
Fix it. Mark the change in the code, above.

13. (8 points) Write a method, firstAndLast, that returns the first and last character in its String parameter pasted together. E.g. if it is passed "iteration" it should return "in", or if it is passed "kart", it should return "kt".

```java
String fandl(String s) {
    return "" + s.charAt(0) + s.charAt(s.length()-1);
}
```

14. (5 points) Say the user is typing a series of chars into a TextField (e.g. when they are trying to guess the letters in a word in hangman). Sometimes, by accident, they type the same letter twice in a row (like by adding it to the TF instead of replacing the previous input); you’d like to ignore that. Add code so that this method will send the process(char) message with the first character in inTF only if it is different from the previous first letter. Hint: local variables are reinitialized each time a method is invoked.

```java
char previous = '*';
private void inTFActionPerformed(java.awt.event.ActionEvent evt) {
    char ch = inTF.charAt(0);
    if (ch == previous) {
        return;
    }
    previous = ch;
    process(ch);
}
```

15. (5 points) Add code to make this send each Innocent in the list an assimilate() message.

```java
ArrayList<Innocent> list = new ArrayList<Innocent>();
...
void assimilateEm() {
    for (Innocent nextI: list)
        nextI.assimilate();
}
```

16. (10 points extra credit) A Star has an income variable with an accessor (see 9). Write a method, mostIncome, which is passed an ArrayList<Star> and returns the Star with the most income.