1. (5 points) Declare an int variable, called anInt, and convert a String, s, to int, store the value in anInt.

```java
int anInt;
anInt = Integer.parseInt(s);
```

2. (4 points) Declare a String variable, s, and convert an int, anInt to a String, and store it in s.

```java
String s;
s = "" + anInt;
```

3. (3 points) What must every identifier in Java start with? A letter.

4. (2 points) What characters are allowed in Java identifiers? Letters, numbers and underscores.

5. (10 points) What are five Java statements we have covered so far? message, return, assignment, empty, block, if, while, for, switch

6. (7 points) Write Java code to sout “Freezing!” if the value of temp is less than 32.

```java
int temp;
if(temp<32)
    sout("Freezing!");
```

7. (5 points) Write a method, named odd, which returns true if its int parameter is an odd number (and false otherwise).

```java
boolean odd(int x) {
    return x%2 != 0;
}
```

8. (9 points) Write a method, named grade, which is passed an int parameter and returns its equivalent letter grade (as a char); 90 or more = A, 80-89 = B, 70-79=C, 60-69=D, <60=F.

```java
boolean grade(int score) {
    switch (score / 10) {
        case 9,10: return 'A';
        case 8: return 'B';
        case 7: return 'C';
        case 6: return 'D';
        default: return 'F';
    }
}
```

9. (5 points) What is the syntax of a while statement? Use BNF or reasonable facsimil of thereof.

```
<while stmt> ::= while (<boolean exp>) <stmt>
```

10. (10 points) Write a loop that will sout the numbers from 1-1000, one per line.

```java
int count=1;
while (count<=1000) {
    sout("" + count);
    count++;
}
```
11. (5 points) Modify your loop so that it only outputs numbers which are even multiples of 13, 17, and 19 (write the changes here and use an arrow to show where they’d go).

```java
insert if (count %13==0 || count %17==0 || count %19==0) before the sout
```

12. (10 points) Here is some broken code:

```java
class Exam {
    int x;
    void doStuff() {
        int x = 17;
        println(x);
    }
    void println() {
        System.out.println("x=" + x);
    }
}
```

When `doStuff()` is executed, to the programmer’s horror, instead of 17, it prints 0 as the value of `x`. What causes this problem? **The declaration of `x` in `doStuff()`**. What is it called? **Shadowing the variable.** How can you fix it? Mark the change in the code, above. **Delete the int before `x=17`.**

13. (10 points) Write a method, reverse, that returns its `String` parameter backwards.

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

14. (5 points) Write a method, `palindrome`, that returns `true` or `false` on the basis of whether its `String` parameter is a palindrome (e.g. noon, pop, and madam are palindromes, cat, evening, and spot are not).

```java
boolean palindrome(String s) {
    return s.equals(reverse(s));
}
```

15. (5 points) Write a method with two parameters, a `String` and a `char`, that will return the number of times the `char` occurs in the `String` (e.g., if the `String` is “elephant”, and the `char`, ‘e’, the correct count is 2).

```java
int countEm(String s, char countMe) {
    int count=0;
    for (int i=0; i<s.length(); i++)
        if (s.charAt(i) == countMe)
            count++;
    return count;
}
```

16. (5 points) Write a for loop that will output the alphabet, one letter per line.

```java
for (char ch='a'; ch <='z'; ch++)
    sout (ch);
```