

Unassessed Programming Tutorial 1

Susan Eisenbach

14 November 2007

1. Translate the following Haskell functions into Java methods.

a)

```
power :: Int -> Int -> Int
-- pre:  the second argument is a non-negative integer
-- post: computes first arg to power of second arg
power x n | n==0 = 1
          | otherwise = x * power x (n-1)
```

b)

```
power1 :: Int -> Int -> Int
-- pre:  the second arg is a non-negative integer
-- post: computes first arg to power of second arg
power1 x n | n==0 = 1
           | n==1 = x
           | n `mod` 2 == 0 = z*z
           | otherwise
             = z*z*x
           where z = power1 x (n `div` 2)
```

In Java ``div`` is `/` and ``mod`` is `%`.

2. Write a method that takes a number between 0 and 9 as an argument and prints the number as a word on the screen. Remember to include the appropriate assertion.

3. Write a method that takes a number between 0 and 9 as a parameter and returns the number as a string from the method. Remember to include the appropriate assertion.

4. Write a method that takes a number between 0 and 9 as an argument and prints on the screen

 "Number green bottles standing on the wall"
(where Number is replaced by the word version of the number). If the numeric parameter is 1 then your method should print bottle rather than bottles. Remember to include the appropriate assertion.

5. This question is for doing after lecture 3.

Write a Java function that calculates factorials using a `for` loop. In Java you can have for loops that go backwards. They are of the form:

```
for (i = 10; i > 0; i--) {  
    loop body  
}
```

Write a program that prints out the song Ten Green Bottles. The words of the song are follows: Ten green bottles standing on the wall.

Ten green bottles standing on the wall.

And if one green bottle should accidentally fall,
there'd be nine green bottles standing on the wall.

...One green bottle standing on the wall.

One green bottle standing on the wall.

And if one green bottle should accidently fall,
there'd be no green bottles standing on the wall.

Answers:

```
int power(int x, int n){
assert (n >= 0);
//post: x^n
    if (n==0) {return 1;}
    else {return x * power(x, n-1);}
}

int power1(int x, int n){
assert (n >= 0);
//post: x^n
    if (n==0) {return 1;}
    else{
        if (n==1) {return x;}
        else{
            int z = power1 (x, n / 2);
            if ((n%2) == 0) {return z*z;}
            else {return z*z*x;}
        }
    }
}

int fact(int n){
assert (n>= 0);
//post: computes n!
    int f = 1; int i;
    for (i=n; i>0; i--){
        f = f*i;
    }
    return f;
}

void printLine(int n){
    println(n + " green bottles standing on the
        wall.");
}

void printRest(int n){
    println("And if one green bottle should accidentally
fall,");

    println("there'd be " + (n-1) + " green bottles standing
on the wall,");
    println("");
}

for (i=10; i>0; i--){
    printLine(i);
    printLine(i);
    printRest(i);
}
fixed:
```

```

void printLine(int n){
assert ((0 <= n) && (n <=9)) : "only numbers between 0 and
9";
    if (n == 1)
        {println(conv(n) + " green bottle standing on
the wall.");}

        else {println(conv(n) + " green bottles standing
on the wall.");}

}

```

and similarly for printRest

```

String conv(int n){
assert ((0 <= n) && (n <=9)) : "only numbers between 0 and
9";
    switch (n){
        case 0: {return "zero";}
        case 1: {return "one";}
        case 2: {return "two";}
        case 3: {return "three";}
        case 4: {return "four";}
        case 5: {return "five";}
        case 6: {return "six";}
        case 7: {return "seven";}
        case 8: {return "eight";}
        case 9: {return "nine";}
        case 10: {return "ten";;}
        default: {return "not possible!";}
    }
}

```