Adrian Paz Barba
Henok Hailemariam
CSCI 3155
Lab 1

2. **Scala Basics: Binding and Scope.** For each the following uses of names, give the line where that name is bound. Briefly explain your reasoning (in no more than 1–2 sentences).

(a) Consider the following Scala code.

```
1  val pi = 3.14
2  def circumference(r: Double): Double = {
3    val pi = 3.14159
4    2.0 * pi * r
5  }
6  def area(r: Double): Double =
7    pi * r * r
```

The use of `pi` at line 4 is bound at which line? The use of `pi` at line 7 is bound at which line?

(a) The use of pi at line 4 is bounded on line
3 **val** pi = 3.14159
since pi was declared again within the scope of def circumference

The use of pi at line 7 is bound on line
1 val pi = 3.14

pi is bound on 1 because it was not declared again within def area

(b) Consider the following Scala code.

```
1  val x = 3
2  def f(x: Int): Int =
3    x match {
4      case 0 => 0
5      case x => {
6        val y = x + 1
7        ({
8           val x = y + 1
9           y
10         } * f(x - 1))
11     }
12   }
13 val y = x + f(x)
```

(b) (i)
The use of x at line 3 is bound by line
2 def f(x: Int): Int =

because the value of x in passed in function f

Adrian Paz Barba
Henok Hailemariam
CSCI 3155
Lab 1

(ii)

The use of x at line 6 is bound by line

5 case x => {

because when line 6 executes it uses the value of x which is determined by the new scope on line 5 which is the case x => {

(iii)

The use of x at line 10 is bound by

5 case x => {

because when line 10 executes it uses the value of x which is determined by the scope on line 5 which is the case x => {

(iv)

The use of x at line 13 is bound by line
1 val x = 3

because it's not part of the scope of function **def f(x: int):**

3. **Scala Basics: Typing**. In the following, I have left off the return type of function g. The body of g is well-typed if we can come up with a valid return type. Is the body of g well-typed?

```
1  def g(x: Int) = {
2    val (a, b) = (1, (x, 3))
3    if (x == 0) (b, 1) else (b, a + 2)
4  }
```

If so, give the return type of g and explain how you determined this type. For this explanation, first, give the types for the names a and b. Then, explain the body expression using the following format:

2 Yes, the body expression of g is well-typed with type ((Int,Int),Int)

(a,b): ((Int,Int),Int) because:
a: (Int, Int)

Adrian Paz Barba
Henok Hailemariam
CSCI 3155
Lab 1

b: Int
if x == 0: ((Int,Int),Int) because:
b: (Int, Int)
1: Int
Else: ((Int,Int),Int) because:
b: (Int, Int)
a + 2: Int