Name	Will Marsh
PSU ID(abc1234)	wem5115
Class (Section)	IST 140.002

Practice Problems 5.2

More on Methods

1) Describe the scope error in the following program and explain how to fix it.

```
import java.util.Scanner;
public class Conversation {
   public static void main(String[] args) {
      Scanner in = new Scanner(System.in);
      System.out.print("What is your first
      name? ");
      String input = in.next();
      System.out.println("Hello " + input);
      System.out.print("How old are you? ");
      int input = in.nextInt();
      input++;
      System.out.println("Next year, you will
      be " + input);
   }
}
```

The variable "input" is already defined and established in line 7 as a string, in line 9, you are re-defining it and re-establishing it as an int. You should rename the variable input to make it more descriptive.

2) For each of the variables in the following program, indicate the scope.

```
public class Sample {
   public static void main(String[] args) {
      int i = 10;
      int b = g(i);
      System.out.println(b + i);
   }

   public static int f(int i){
      int n = 0;
      while (n * n <= i ){n++;}
      return n - 1;
   }

   public static int g(int a){
      int b = 0;
      for(int n = 0; n < a; n++){
         int i = f(n);
         b = b + i;
      }
      return b;
   }
}
```

For this one, int 'i' is set to equal 10 until line 6, where it loses value. Same thing with int 'b'.

For this one, you re-establish int i and int n (setting n = 0). They are both established until line 12, where they lose value.

For this one, you establish int 'a', int 'b', and int 'n'. Int 'n' and int 'i' are only established between lines 16-19. Int 'a' is established from line 14-19. You would not be able to return 'b' because it is outside the established region.

Name	Will Marsh
PSU ID(abc1234)	wem5115
Class (Section)	IST 140.002

3) Consider the following method that is intended to swap the values of two integers:

```
public static void falseSwap(int a, int b){
   int temp = a;
   a = b;
   b = temp;
}

public static void main(String[] args) {
   int x = 3;
   int y = 4;
   falseSwap(x, y);
   System.out.println(x = " " + y);
}
```

Why doesn't the falseSwap method swap the contents of x and y?

Because there is no overlap between the first section and the second. The values are just kept their same value and never switch as they were intended to.