Homework 1

Blair Sibu

Problem 1. Given two files,
(1) What happens when you compile and execute them?

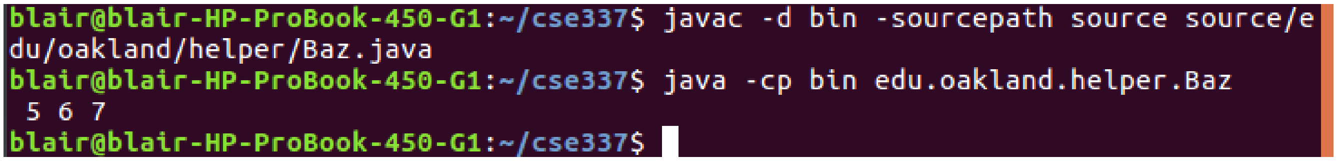
When I try to compile them two errors occur. The first error is that “a” cannot be accessed from outside the package because its access modifier is set to default. The second error is similar but pertains to the variable “b.” b cannot be accessed outside the package because it has protected access, and protected access is only package level access. Since both files are in different packages b cannot be reached.

(2) Make any changes necessary to successfully compile and run this Java program.

To fix this I made Baz a subclass of Foo to be able to access the protected variable outside the package, and then I changed the modifier for “a” to public so that I could reach it from outside the package. There was also the necessary change of adding the package and import statements.

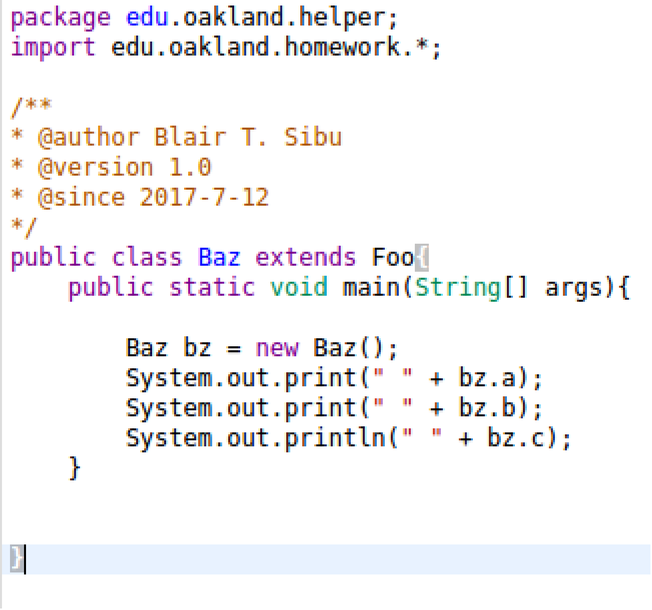
See code for details.

Screenshot:



```
blair@blair-HP-ProBook-450-G1:~/cse337$ javac -d bin -sourcepath source source/e
du/oakland/helper/Baz.java
blair@blair-HP-ProBook-450-G1:~/cse337$ java -cp bin edu.oakland.helper.Baz
 5 6 7
blair@blair-HP-ProBook-450-G1:~/cse337$ 
```

Code:

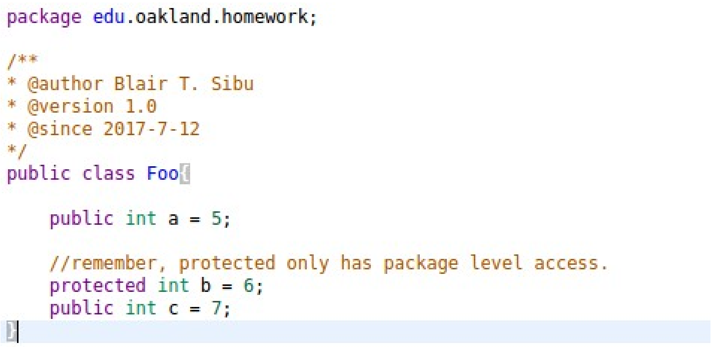


```java
package edu.oakland.helper;
import edu.oakland.homework.*;

/**
* @author Blair T. Sibu
* @version 1.0
* @since 2017-7-12
*/
public class Baz extends Foo{
    public static void main(String[] args){

        Baz bz = new Baz();
        System.out.print(" " + bz.a);
        System.out.print(" " + bz.b);
        System.out.println(" " + bz.c);
    }


}
```



```java
package edu.oakland.homework;

/**
* @author Blair T. Sibu
* @version 1.0
* @since 2017-7-12
*/
public class Foo{

    public int a = 5;

    //remember, protected only has package level access.
    protected int b = 6;
    public int c = 7;
}
```

Homework 1

Blair Sibu

Problem 2. Given:
(1) What is the result when you compile and execute?

When I try to compile it I receive four errors. The first error, and possibly the more obvious, is that we're trying to pass a short to the method countGold but that method only accepts two parameters that are both integers. Changing the first parameter to short fixes this error. The other errors deal with access modifiers. The first is that the class Hobbit had a default access modifier and so it couldn't be reached from outside the package. We can fix this by changing it to public. The method inside of Hobbit, countGold, must also be changed to public. When that's all said and done it still won't compile because there's still the problem of having to write and import statement so that it can properly inherit the class Hobbit. To be able to access the method countGold from Frodo, we need to make an instance of an object for it.

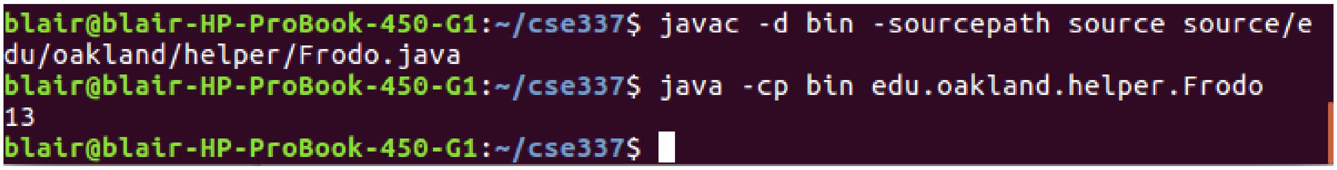
See Code for details

(2) Make any changes necessary to make these files compile and execute successfully.

There was also of course the necessary change of adding the package statement.

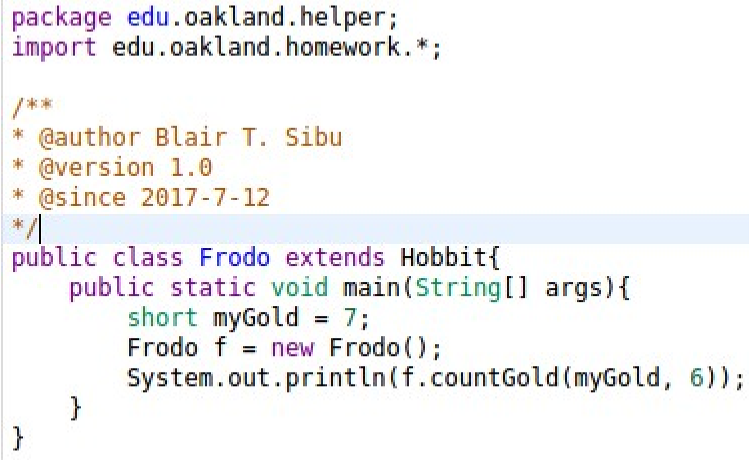
See (1) for more necessary changes

Screenshot:



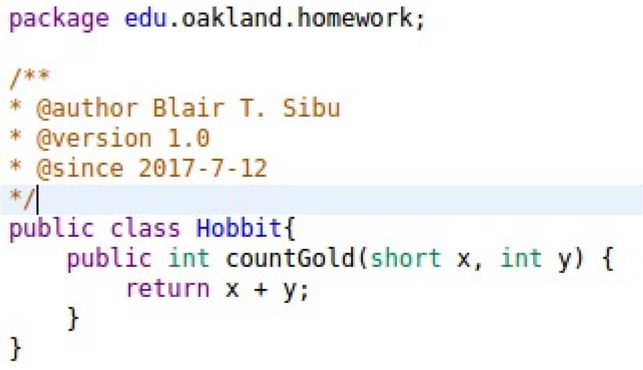
```
blair@blair-HP-ProBook-450-G1:~/cse337$ javac -d bin -sourcepath source source/e
du/oakland/helper/Frodo.java
blair@blair-HP-ProBook-450-G1:~/cse337$ java -cp bin edu.oakland.helper.Frodo
13
blair@blair-HP-ProBook-450-G1:~/cse337$ 
```

Code:



```java
package edu.oakland.helper;
import edu.oakland.homework.*;

/**
* @author Blair T. Sibu
* @version 1.0
* @since 2017-7-12
*/
public class Frodo extends Hobbit{
    public static void main(String[] args){
        short myGold = 7;
        Frodo f = new Frodo();
        System.out.println(f.countGold(myGold, 6));
    }
}
```



```java
package edu.oakland.homework;

/**
* @author Blair T. Sibu
* @version 1.0
* @since 2017-7-12
*/
public class Hobbit{
    public int countGold(short x, int y) {
        return x + y;
    }
}
```

Homework 1

Blair Sibu

Problem 3. Given:

(1) What is the result when you compile and execute them?

Upon initial compile we of course get an error because there’s no import statement. After adding the import statement we get another error about Building not being public. We can’t inherit anything if Building is not public. We must also make the constructors of Building public for the same reason. I also made the constructors of House public because it almost seems like this class would be inherited by something else, e.g. Blairs_House. Note that I also changed to last “print” to a “println” to start the prompt on a new line, it makes the information more pleasing to look at without the current directory getting in the way.

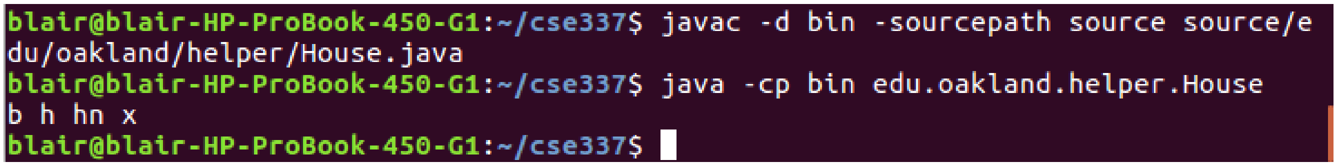
See Code for details

(2) Make any changes necessary to make these files compile and execute successfully.

There was also of course the necessary change of adding the package statement.

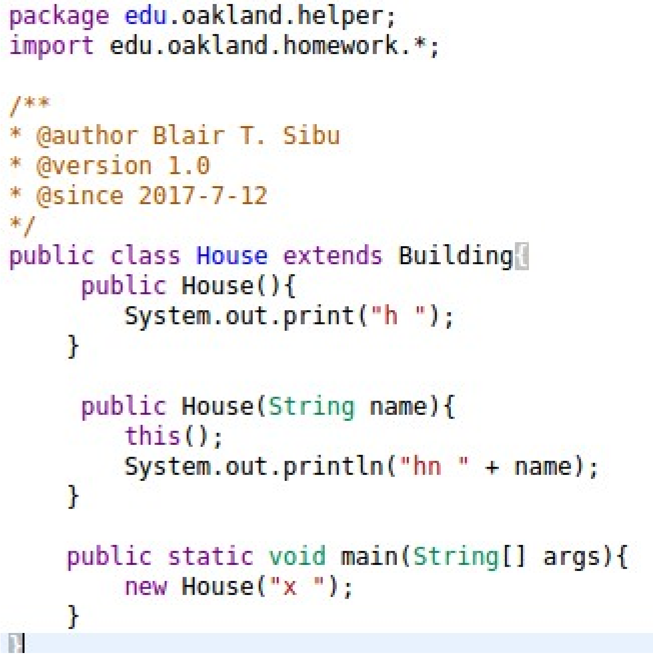
See (1) for more necessary changes

Screenshot:



```
blair@blair-HP-ProBook-450-G1:~/cse337$ javac -d bin -sourcepath source source/e
du/oakland/helper/House.java
blair@blair-HP-ProBook-450-G1:~/cse337$ java -cp bin edu.oakland.helper.House
b h hn x
blair@blair-HP-ProBook-450-G1:~/cse337$ 
```

Code:

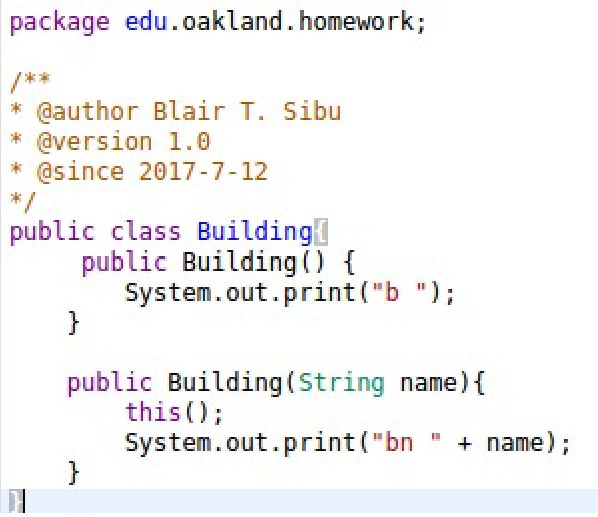


```java
package edu.oakland.helper;
import edu.oakland.homework.*;

/**
* @author Blair T. Sibu
* @version 1.0
* @since 2017-7-12
*/
public class House extends Building{
      public House(){
         System.out.print("h ");
     }

      public House(String name){
         this();
         System.out.println("hn " + name);
     }

     public static void main(String[] args){
         new House("x ");
     }
}
```



```java
package edu.oakland.homework;

/**
* @author Blair T. Sibu
* @version 1.0
* @since 2017-7-12
*/
public class Building{
      public Building() {
         System.out.print("b ");
     }

     public Building(String name){
         this();
         System.out.print("bn " + name);
     }
}
```

Homework 1

Blair Sibu

Problem 4. Given:
(1) What is the result when you compile and execute?

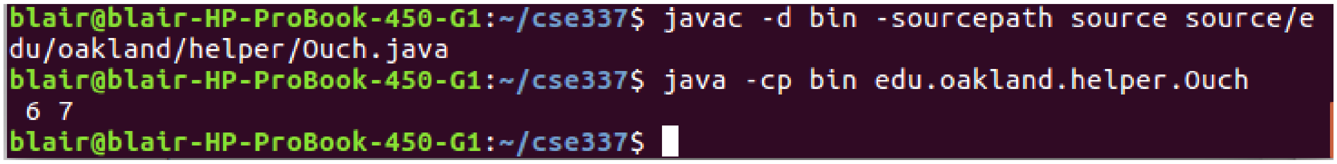
The only error that I received when trying to compile is that for the "for" loop, since the variable "ouch" was already defined as an int for the "go" method we don't need to redefine it in the "for" loop.

(2) Make any changes necessary to make this file compile and execute successfully.

To fix this error simply remove the int primitive definition from the variable "ouch" within the "for" loop. I've also included the package statement.

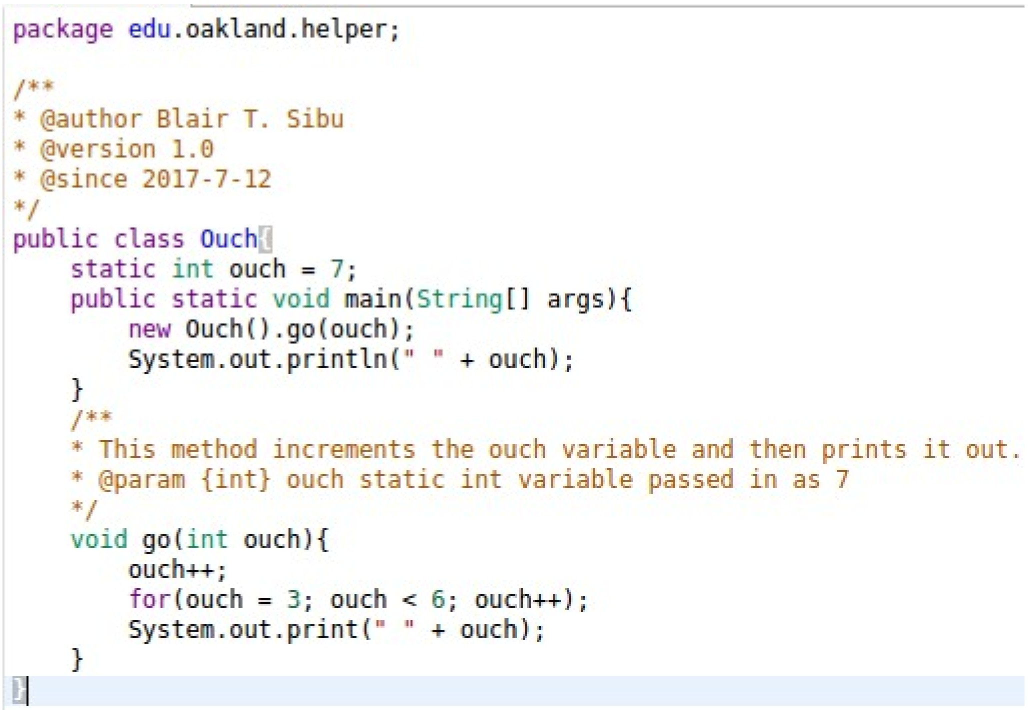
See Code for details

Screenshot:



```
blair@blair-HP-ProBook-450-G1:~/cse337$ javac -d bin -sourcepath source source/e
du/oakland/helper/Ouch.java
blair@blair-HP-ProBook-450-G1:~/cse337$ java -cp bin edu.oakland.helper.Ouch
 6 7
blair@blair-HP-ProBook-450-G1:~/cse337$ 
```

Code:



```
package edu.oakland.helper;

/**
* @author Blair T. Sibu
* @version 1.0
* @since 2017-7-12
*/
public class Ouch{
    static int ouch = 7;
    public static void main(String[] args){
        new Ouch().go(ouch);
        System.out.println(" " + ouch);
    }
    /**
    * This method increments the ouch variable and then prints it out.
    * @param {int} ouch static int variable passed in as 7
    */
    void go(int ouch){
        ouch++;
        for(ouch = 3; ouch < 6; ouch++);
        System.out.print(" " + ouch);
    }
}
```

Homework 1

Blair Sibu

Problem 5. Given:

(1) What is the result when you compile and execute?

When I tried to compile it I received one error. The error said, “variable x might not have been initialized.” This was inside of the method, “go1” when we try to pass the variable, “x” to the method, “go2.” The method isn’t being passed the variable from the main method. I decided that I wanted to do as minimal work as possible on this one and instead of using the x variable in the main and having to add in another line of code to be able to access that variable, I initialized the x variable to zero inside the go1 method making it a local variable. The only requirements were to make necessary changes for a successful compile.

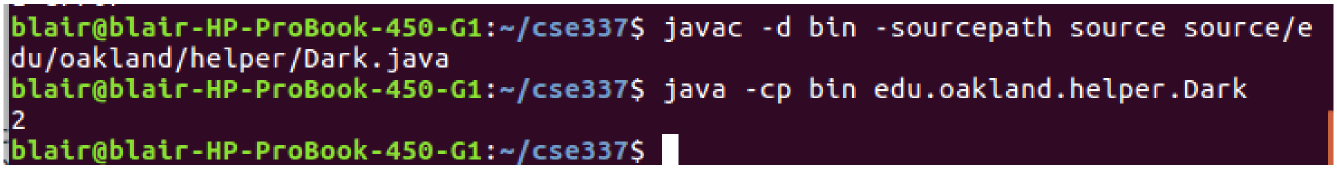
See Code for details.

(2) Make any changes necessary to make these files compile and execute successfully.

There was also of course the necessary change of adding the package statement.

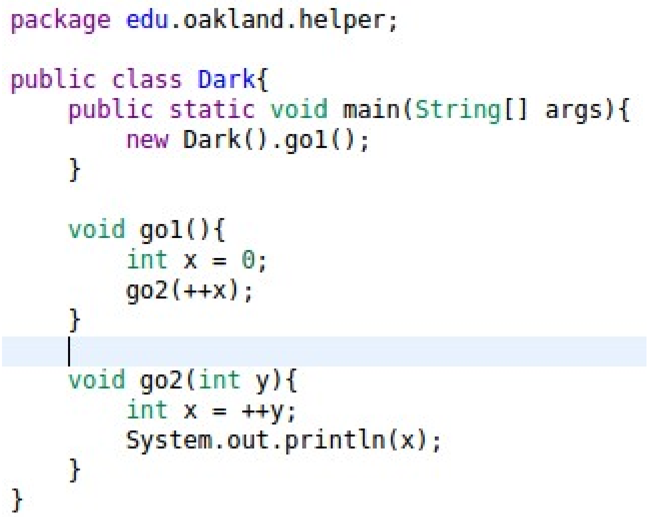
See (1) for more necessary changes

Screenshot:



```
blair@blair-HP-ProBook-450-G1:~/cse337$ javac -d bin -sourcepath source source/e
du/oakland/helper/Dark.java
blair@blair-HP-ProBook-450-G1:~/cse337$ java -cp bin edu.oakland.helper.Dark
2
blair@blair-HP-ProBook-450-G1:~/cse337$
```

Code:



```java
package edu.oakland.helper;

public class Dark{
    public static void main(String[] args){
        new Dark().go1();
    }

    void go1(){
        int x = 0;
        go2(++x);
    }

    void go2(int y){
        int x = ++y;
        System.out.println(x);
    }
}
```