File I/O Programs

Things that need fixing:

1. `public static void main(String[] args) throws IOException {`
   No. You need to handle the exception properly with a try-catch

2. ```
   import java.util.*;
   import java.io.*;
   ```
   Don't do this. You need to specify which imports you are actually using.
   Eclipse has an option "Source / organize imports" that fixes this.

3. `String filename = "words_alpha.txt";`
   Always put the filename as a variable so that you can refer to it later.

4. ```
   catch (FileNotFoundException e1) {
      e1.printStackTrace();
   }
   ```
   If there is any way to provide more explanation of an error, you need to do so.
   FileNotFound is an easy one to fix. (see below)

5. ```
   catch (FileNotFoundException e1) {
     System.out.printf("File \"%s\" was not found. Exiting.",filename);
     System.exit(0);
   }
   ```
   You have to decide if you need to stop the program at this point or not.
   Eclipse cannot decide this for you.

6. ```
   catch (Exception e) {
      e.printStackTrace();
   }
   ```
   Never catch "Exception e". It is too general. Be as specific as possible.

7. Organize your program so that it makes sense.
   This means that you'll need to move things into a try {} block or declare a variable outside a try{} block so that it can be accessed later.
   Don't rely on Eclipse to do this. It can't make your program efficient.

8. There is rarely any reason to create a whole lot of i/o objects
   `BufferedReader br = new BufferedReader( new FileReader( new File"words_alpha.txt")));`
   Notice that we only have one variable: **br** . We don't have a variable for the File or FileReader objects.

9. FYI: FileReader can take a filename as a string and doesn't need a File object!
   ```
   new FileReader(new File"words_alpha.txt")));
   new FileReader("words_alpha.txt"));
   ```
   I think Scanner needs a File object.

10. You can catch multiple errors:

```
try {
	BufferedReader br=new BufferedReader(
		new FileReader("words_alpha.txt"));
	for (int i=0;i<15;i++) {
		String b=br.readLine();
		System.out.println(b);
	}
	br.close();
} catch (FileNotFoundException e) {
	 System.out.printf("File \"%s\" was not found. Exiting.",filename);
	System.exit(0);
} catch (IOException e) {
	e.printStackTrace();
	System.exit(0);
}
```

11. Put the most specific exceptions first.

    Since `FileNotFoundException extends IOException` it should be first.

    This means you have to check to see if one of the exceptions extends the other.

12. ***Other strange things:***

    Putting a try-catch inside a for loop is probably backwards. Put the forloop inside the try-catch.

```
for (int i = 0; i < 15; i++)
 {
	String word2 = null;
	try {
		word2 = br.readLine();
	} catch (IOException e) {
		e.printStackTrace();
	}
	System.out.print(word2 + " ");
}
```

13. When you read from a file with Scanner or BufferedReader, you **have to** close the file when you're done with it.

```
sc = new Scanner(new File(filename));
...
sc.close();
```

14. DO NOT EVER put an absolute path name unless you have a very good reason to do this.

```
File input = new File("X:\\My Drive\\Eclipse WorkSpace 2\\ICS4U1\\src\\
words_alpha.txt");
```

    This will not work on anyone else's computer.

THE END (for now)