

## CS 051G Homework Laboratory # 11

### Back to the Future

**Objective:** To gain experience using Java

---

**The Scenario.** For this problem you will model a sticky carpet that collects balls of dust.

- The program begins with a FilledRect that represents the carpet.
- On a mouse press a black FilledOval (the dustball) that is centered on the point that the user clicks on is created.
- On a mouse drag the dustball moves around as expected.
- On a mouse release the dustball turns blue and can no longer be moved.
- However, if at any point the dust overlaps the carpet (when the dust is initially created or while it is being dragged), the dustball turns red and can no longer be moved.

**Wait a minute, didn't we already do this lab? Yes, but now you are doing it in Java instead of Grace. However, you get to to use your Grace solution to guide your Java implementation!**

**Using Eclipse to write Java programs** You must follow the following instructions carefully as they are different from those that worked in Grace. First, copy the folder Lab11–StickyCarpetRedux from the folder cs051G/labs. Copy this into your CSC51GWorkspace folder. Don't neglect to copy this folder. If you don't copy it you won't be able to save your program and may lose it.

Now click on the Eclipse item in the menu bar at the bottom of the screen. When it comes up, it may ask you which folder to use at the workspace. Tell it to use your CSC51GWorkspace folder.

For the very first lab, you are going to need to do some configuration of Eclipse to make it work better for the rest of the term. Here is what you need to do:

1. From the eclipse menu item on the tool bar at the top of the screen, select "Preferences...". Along the left edge of the pop-up window, click on the triangle next to the word "Java". That will open a submenu. Click on the triangle next to "Build Path" and then click on "Classpath Variables".
2. On the right side of that window, click on "New...". A new window will pop up. In the "Name" field type "OBJECTDRAW" (in all caps, but without the quotes). In the "Path" field type `/common/cs/cs051/objectdraw/objectdraw.jar` This must be exact or you will not be able to run objectdraw programs. (Alternatively, you can click on the "File..." button and navigate to your CS51Workspace folder and then objectdraw and the objectdraw.jar file.)

Once this is done, click OK a few times until you are back on the preferences window.

3. We need you to do one last thing to eliminate an annoying warning message. In the same left pane of the Preferences window as before, under Java, click on the triangle next to "Compiler" and then on "Errors/Warnings". On the right side of the window you will see categories of issues. Click on the triangle next to "Potential programming problems" and scroll down to "Serializable class without serialVersionUID" and select "Ignore" as the option. Click OK and you are done with this one time setting of preferences and ready to work!

In the Eclipse window, pull down the "File" menu and select "Import..." from near the bottom of the menu. Select "Existing Project into Workspace" and click the "Next" button. [You may have to click on the triangle next to "General" to see the "Existing Project ..." entry.] Now, click the "Browse" button to the right of the "Select root directory" entry, and navigate to your CSC51GWorkspace folder. (Hint: double-click on "Desktop" if you seem lost.). Double click on CS51Workspace and then click on "Lab11-StickyCarpet". Push "OK." Then push the "Finish" button.

The eclipse workspace window may be obscured by a welcome screen. If so, click on the "x" to remove it.

On the left side of the Eclipse window, in an area called the "Package Explorer" you should now see an entry for "StickyCarpet". Click once on the triangle to the left of "StickyCarpet" and then once on the triangle to the left of "(default package)". Then double-click on "StickyCarpet.java".

This window has three main regions. On the left, it displays the names of files and their contents in an outline form. The large area on the top right side of the screen displays the contents of "StickyCarpet.java" file. You will edit your files here.

The simplest way to compile and run your program is using the white triangle in the green circle above your program file. The first time you want to run your program you should make sure that the main program (the one extending `WindowController`) is open and showing in the editor window. Then use the white triangle as a popup menu. When you select the menu, one of the options should be `Run as ...`. Select that and follow the walking menu to `Java application`. Once you have done this, the next time you can just click on the button and it will compile your program.

This "Run" button will first compile your program and then, assuming there are no errors, will begin executing it. One nice thing about Java is that it will statically type check your program to see if you are doing something that makes no sense from the point of view of the types you have declared. You will find lots of errors that way, and the system will print all of them that it can detect rather than being limited to one like the Grace compiler.

Compile-time errors will show up as red X's in the left margin of the editor window. If you let the mouse hover over the x, it will show a brief error message. You will often get even more information if you click on it.

OK, that should be enough to get you going in using Eclipse. Here are some last tips to remind you of the differences between Java and Grace:

1. There are minor differences in the names of methods in the Java `objectdraw` library compared to Grace. Keep the `objectdraw` API documentation open at <http://www.cs.pomona.edu/classes/cs051/handouts/objectdraw-api.html>.
2. You must put a semicolon at the end of lines in Java. However, never put a semicolon before a `{` or after a `}`.
3. Parameterless method requests must include `()` even though there are no parameters.
4. Every Java class must be in a separate file.
5. Every instance variable, constant, and method must have a visibility annotation. For now, use one of `private` and `protected`. Do not put these in front of local variables.
6. Types are written before the identifiers in declarations. E.g., `int x = 5;`.
7. Assignments to variables are written with `=` instead of `:=`.
8. Constants are written in all caps and are declared as `static final` if outside of methods. Inside methods (local constants), just declare them as `final`.

9. Uninitialized instance variables are given the value 0 (if a primitive type) or null (if an object type). If you forget to initialize an instance variable you will get a “null pointer error”. Local variables are not automatically initialized. You must make sure to initialize them.
10. In Java we use keyword `this` in place of `self`.

You do not have to bring in a design for this week’s lab, but you should bring in a(n electronic) copy of your StickyCarpet program in Grace as you will use that to guide your Java solution.

**Getting Started** The starter project contains `src` and `bin` directories. These won’t even be visible from inside Eclipse. If you did all the copying directed above, you will just need to work on the StickyCarpet class in the eclipse editing window. Be sure to save your program each time you want to compile and run it.

**Submitting Your Work** Before submitting your work, make sure that your program includes your name in the comment heading up the class. Also, before turning in your work, be sure to double check both its logical organization and your style of presentation. Make your code as clear as possible and include appropriate comments describing major sections of code and declarations. Use the Format command in the Source menu to make sure your indentation is all consistent.

**Turning in your program** Your program is due at 11p.m. on Tuesday or Wednesday evening (depending on your lab section). However, the goal is to finish it during the lab so you can focus on your test program.

Turning in your program is a little more elaborate in Eclipse and Java than it was for Grace.

- First, return to Eclipse and make sure you included your name and course number in a comment at the start of each of the Java classes.
- Next, select the “StickyCarpet” project in the Package Explorer panel on the left side of the Eclipse window.
- Now, select “Export” from the “File menu.
- Select “File system” in the dialog box and click next.
- Make sure the folder with your project has a check mark next to it. (That will make all the files on the right have checks too!)
- Click the “Browse” button next to the “To directory:” entry.
- Double click on “Desktop”.
- Click the “Finish” button. It will create a new folder on your desktop.
- Quit Eclipse.
- Click on the new folder’s name and type in a name for the folder that identifies you and the lab you are working on, such as “*Yourname-Lab11*” and then click “OK”. (Dashes in names are OK, but don’t include spaces or periods.)
- Open the cs051G folder icon to find the “dropbox” folder.

- Drag your new folder into the dropbox folder. When you do this, the computer will warn you that you will not be able to look at this folder. That is fine. Just click “OK”.
- The new folder will still show up on your desktop. Drag it into the trash both to save space and to keep anyone from copying it. (Files in your CSC 051 folder are protected so that others cannot read them.) Select “empty trash” from the file menu to completely delete those files. Do NOT throw away the version of the program in your workspace. You should save that so you can study it later.

If you should accidentally turn in a bad version of your program, you may drag another copy in as long as you change the name to be slightly different from the one you used before (e.g. Jane Doe - lab 11a). The new name should make it clear which is the newer version.

Table 1: Grading Guidelines

Value	Feature
	<b>Design (2 free pts!)</b>
	<b>Readability (6 pts total)</b>
2 pts.	Descriptive comments
1 pts.	Good names
2 pts.	Good use of constants
1 pts.	Appropriate formatting
	<b>Code Quality (4 pts total)</b>
2 pts.	Good use of boolean expressions
1 pt.	Not doing more work than necessary
1 pt.	Using most appropriate methods
	<b>Correctness (8 pts total)</b>
2 pts.	Carpet drawn correctly at startup
2 pts.	Dustballs can appear and can be moved
2 pts.	dustballs turn correct color when released
2 pts.	Dustballs stick to carpet