Computer Science Department

Mount Holyoke College

CS 101 - Problem Solving and Structured Programming
Test Program 2 - Desktop
Due: December 10, 5:00 PM
THIS IS A FIRM DEADLINE!

This assignment is a take-home test. As such, the honor code rules are different for this assignment than for regular labs. Here are the rules:

- The only person you may consult with is Professor Lerner. You may not consult with each other, with our TA or lab instructor, or with anyone outside of this class.
- You may consult your notes, your book, your previous assignments, and the course web-site. You may download and examine all examples available on the course website. You may not otherwise use the Internet to help complete this test.

If you have any questions about whether a particular activity is in violation of the honor code, please do not hesitate to ask. Also, please do not hesitate to ask me questions about the exam. It will never hurt! I reserve the right to not answer a question, but, in general, you should expect asking questions to be helpful.

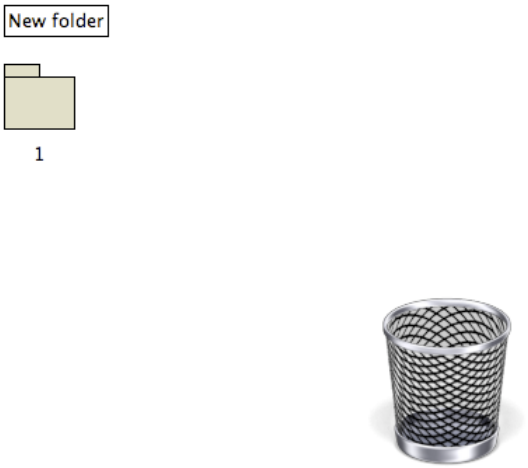
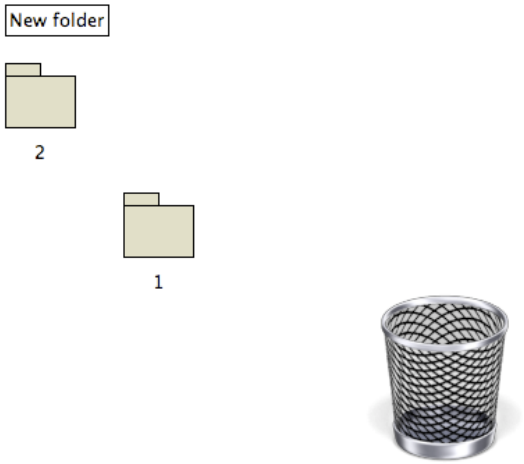
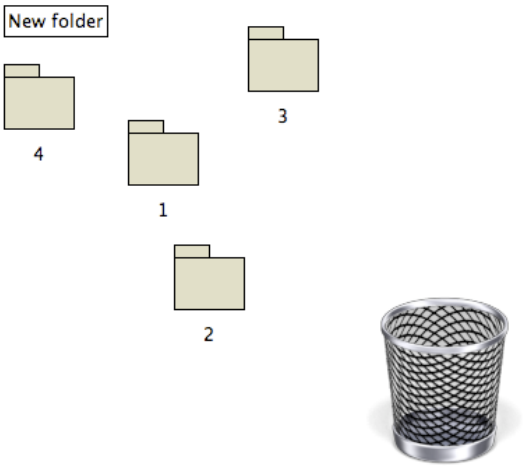
By submitting your solution to this exam via Ella, you are asserting that you have adhered to these honor code rules.

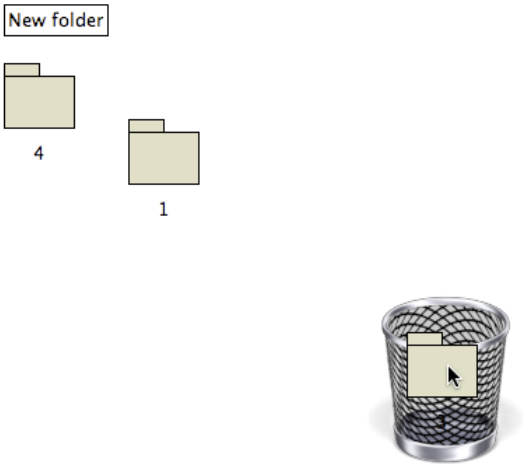
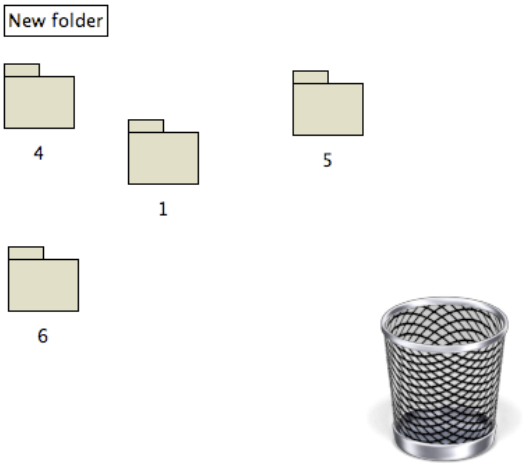
Introduction to the Assignment

For this test program, you will create a program that contains a few features available on a desktop. In particular, you will be able to create folders, drag folders around and delete folders by dragging them to the trash.

Here are some snapshots from this program:

New folder 	This is the original display. It contains a button used to create new folders and a trash can

	Here is the display after the user clicked the New Folder button. A folder is created and given the name 1.
	Now the user has dragged folder 1 out of the way and created a 2nd folder. It is ok for folders to always be created in the same location.
	Now, we have 4 folders, all dragged to different locations. Names are created by counting the folders as they are constructed and using the count for the name. Folder 1, Folder 2, etc. (You have not learned how to do keyboard input to allow a user to name the folders herself.)

	Here the user has already deleted 1 folder by dragging it to the trash and is about to delete a 2nd folder.
	After deleting some folders, the user returns to creating more folders.

The Program Design

You will write 4 classes for this program: Desktop, Button, FolderCollection and Folder. In addition, you will use the Picture class that you used in ImageManipulator. As before, you should not change the Picture class at all.

The Desktop class extends WindowController. It is responsible for defining the begin method and the event handlers.

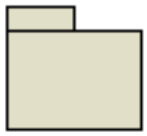
- begin method - This method should draw the starting screen and do any other initialization your program requires. The trash can image is in a file called trashempty.png, which is available for download from the course web site and from Ella.
- onMouseClick method - This method should handle clicks on the New Folder button. Count the folders as you create them so that you can give each folder a unique name.
- In addition, you need to define the mouse handling methods needed to drag the folders around. Remember to delete folders when they are dragged to the trash.

The FolderCollection class manages the entire collection of folders. It provides:

- A constructor that creates an empty array large enough to hold 10 folders
- A method to add a folder to the collection. This method should do nothing if the collection is already full.

- A method to find a folder at a particular location
- A method to remove a particular folder from the collection. When the user deletes a folder, this will mean there is now an empty slot into which a new folder cold be inserted if the user creates a new folder.

The Folder class defines an individual folder. A Folder should look like the image on the right. It consists of 2 colored filled rectangle with 2 framed rectangles drawn over them to give the border. The folder's name should be centered under the folder. The folder can be any color as long as the borders are visible. The Folder class includes:



- A constructor to create a new folder and display it.
- Methods needed to support dragging folders. A folder should only drag if the user clicks on the rectangles, not the text.
- A removeFromCanvas method to remove the folder from the canvas.

The Button class should be familiar to you from previous labs.

Writing the Program

There are many orders in which you might go about writing this program. Here is one.

Step 1: Create the starting display, but have it do nothing.

Step 2: Clicking New Folder button should create a new folder at a fixed location.

Step 3: Drag last folder created.

Step 4: Drag any folder created. If user clicks in an overlapping area, drag topmost folder.

Step 5: If user drags a folder to the trash, remove the folder from the collection and from the display.

Notes: If the collection becomes full, it is ok to create more folders. You just won't be able to drag them. If the user drags a folder to trash, that should free up room so another new folder could be created and dragged.

Grading

10	Step 1: Creating the starting display
10	Step 2: Implementing new folder button
10	Step 3: Dragging a folder
10	Step 4: Selecting a folder to drag
10	Step 5: Deleting folders
10	Comments
5	Constant declarations

5	Indentation
10	Variable names
5	Instance variables / local variables / parameters
5	If statements / loops
5	Method design
5	Arrays

LATE PENALTY: You will lose 10 points for each day that your submission is late.

Turning in Your Work

Go to ella. Click on COMSC 101 in the toolbar menu across the top. Then click on Assignments in the left column. Click the submit link for this assignment. Click the Add Attachment button. Use the Browse… button to upload a local file. You should submit your Java files, but not the other files created by BlueJ. Click Continue. Then click the Submit button.

Once you have submitted a solution to an assignment, Ella does not allow you to resubmit your assignment. In order to allow you to submit a revision, I will create "extra" assignments. So, in Ella, you should see assignments something like this:

- Test Program 2
- Test Program 2, revision 1
- Test Program 2, revision 2

You only need to submit a revision if you would like to change your solution after submitting it. Please ask if you are confused about this.