

CS U231 Reading Assignments

The goal of these assignments is to have you intensely read the Java code and web code for the *PicturePages* program and the web code for the CS U231 web site. This will enable you to begin to understand how Java programs are written and how web sites are constructed. It is no accident that the purpose of the *PicturePages* program is to automatically construct a web site for a set of images. This allows you to understand the core structure of a web page as you understand how such a page might be generated automatically in Java.

As you do these assignments, refer to the handouts distributed in class.

Hints

As you look at the Java examples, take notes about:

- What statements define, specify, or update *data elements*?
- What statements specify *behavior* within *functions / methods*?
- What is the meaning, as best as you can ascertain, of the various constructs?
- What relationships do you notice?
- Where does the data come from, how is it transformed, and where does it go?
- *What questions do you have and what aspects remain unsettled or puzzling?*

You may take these notes directly on your copy of the handouts or you may use sticky post-it notes so that when questions are resolved then the notes can be thrown away.

As you look at the web examples, ask similar questions about what is the effect of the *tags* (the items enclosed in angle brackets <...>). If you look at the corresponding *cascading style sheet* file (with the file extension .css), you will see that some of the standard tags are modified by the definitions in the file.

PicturePages

Data elements

The *PicturePages.java* file has 4 data elements that are *variables accessible to all methods*, that is, that are enclosed in the class as a whole and that are defined or may change as the program proceeds. These variables are:

- **directory**
- **pagetitle**
- **files**
- **length**

Using a yellow highlighter, mark everywhere in the Java file where these variables occur. Then note where each variable is defined or introduced, where it is set via assignment (=), and where is used but not changed by assignment.

The Java file also has numerous *constants* marked with the keyword **final**.

Functions / Methods

The *PicturePages.java* file has 14 functions / methods. You may consider *function* and *method* to be synonyms. The functions / methods are:

- **makePictureWebSite**
- **readImageFileNames**
- **refineFileName**
- **createPictureWebSite**
- **createPictureWebPage**
- **appendSideLinks**
- **linkTo**
- **createWebPageName**
- **createWebPagePathName**
- **threeChars**
- **makePictureLink**
- **printRawFileNames**
- **printRefinedFileNames**
- **makePictureWebSiteGUI**

Again, using a highlighter (perhaps of a second color), mark where each function / method is defined.

To get a better handle on how the functions work, it is useful to understand what functions any given function *uses* or *calls*. We say that *function A calls function B* if B appears in the body of the definition of function A. For example, the function name **makePictureWebSite** appears in the body of **makePictureWebSiteGUI** so **makePictureWebSiteGUI** *calls* **makePictureWebSite**.

Again, using a highlighter (perhaps of a third color), mark each function name that appears in the body of a function definition. You will then see what functions the defined function calls.

To understand the relationship between functions, draw a *function call graph*. Start with the function **makePictureWebSiteGUI**. List this name at the top of a sheet of paper. Then find out what functions it calls. You should find only one in this case, **makePictureWebSite**. Write this name below and draw an arrow from the first function to this function. Now, repeat the process with **makePictureWebSite**. You find that it calls two functions, **readImageFileNames** and **createPictureWebSite**. Write down these names and draw appropriate arrows. Continue this process until you have finished and can go no further. You will have the *function call graph*.

Note that you will not be able to reach 2 of the 14 functions. These functions represent test code that is not used in the final program.

Note also that there are no loops formed by the arrows in the function call graph. This is because *the program is structurally simple and has no recursion*. More subtle programs will use recursion and will have such loops in the function call graph. You will learn to use recursion later and you will love it. The course CS U211 will also discuss recursion in depth.

Now ask the following questions (and more that you may invent yourself):

- Where and how is the name of the directory that hold the pictures entered by the user?
- Where and how is the common title of each web page entered by the user?
- Where and how are the names of the picture files gathered?
- Each picture in the picture directory corresponds to one web page since the site shows one picture per web page. What method orchestrates the creation of the web page for a particular picture?
- Every single character that appears in one of the web pages is generated by some constant or piece of code in the program. Can you track where the different chunks of web page text come from? Furthermore, which chunks of web page text are the same for all web page files? Where is the web page text that is *not constant* on all web pages constructed? How is this *not constant* text managed?
- Where and how does the accumulated web page text actually get saved to one of the web page files?
- What naming convention is used for the web page file names? What part of the program makes sure that this convention is followed uniformly? What is the special role of the web page file *index.htm*? How is linking *back* to the previous web page and forward to the *next* web page managed? Ditto, for *first* and *last*?
- The small *cascading style sheet* file sits outside of the Java program and outside of each of the web page files. How does it have any influence?

The CS U231 web Page

Do a similar analysis of what is happening in the *html* and *cascading style sheet* that is used to create the CS U231 web site page. Ask yourself some questions. Figure out what is going on.