JOURNEY TO MARS

There and Back Again

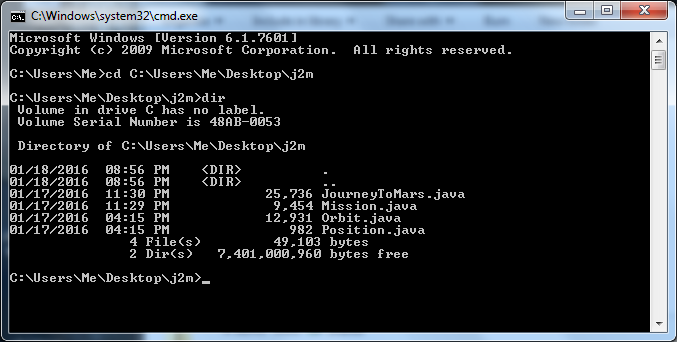
This assignment will introduce you to how software projects are (often) run in the *real* world. Specifically, you will leverage your current programming skills to understand, troubleshoot, and modify an existing piece of software (ah, yes…the joys of working with someone *else’s* code!). Over the course of this assignment, you will be given a series of tasks. Many of these tasks will originate from requests by the *end user*. As you will soon see, the *end user* is mostly concerned with “what”, and less so with “how” (or “why”). Hence, many of these tasks will lack sufficient technical detail, and it will be up to you to fill in the gaps. Take your time, *think* before you code, and don’t be afraid to ask questions. Good luck!

**Level 1**

Congratulations! Fresh out of college, computer science degree in hand, you have landed your dream job at the NASA Jet Propulsion Laboratory (JPL) in sunny Pasadena, CA. Not that you would know much about the sun, though. Since your arrival two weeks ago, you have been working in a dark, window-less, basement-level “cube farm”. Signs of life are scarce so far, but they did place you close enough to the restroom to occasionally hear the toilets flush. You know there must be *someone* out there…at least you hope so!

Your suspicions are confirmed that morning when your boss suddenly peers above your cubicle wall, throws over a stack of hand-written notes and a floppy disk (yes, a floppy!), and makes a hasty retreat (back to a bright and cheery corner office, no doubt!). You hear him in the distance, like a passing train, bellow, “Looks like we’re sending humans to Mars…again! Review this orbit calculator from 2004 and have an SDD on my desk by COB today!”

SDD? COB? **SMH!!!!** A quick Google search reveals that an SDD is a “Software Design Document” and COB means “Close of Business”. Uh-oh, there’s not much time, you’d better get crackin’! Not surprisingly in this day and age, your workstation does not have a floppy drive; but you find an old clunker in the abandoned cube next to yours. You brush the cobwebs aside, boot up the old dinosaur, and *feed* it the floppy. This is what you see:



**Exercise 1A:** Review the four Java files shown in the previous figure. In the space below, list each associated class (e.g. JourneyToMars, Mission, etc.) and include one sentence that describes what function that class performs. *HINT: Don’t waste too much time trying to understand everything about a given class. Look at the comments included in each Java file to get a general idea of what the class does.*

**Exercise 1B**: Draw a simple diagram that illustrates how the four classes are related. For instance if Class A creates an instance of Class B (e.g., *ClassB classb = new ClassB()*), draw blocks to represent the instances of Class A and Class B. Label each block with the name of the given instance (e.g., *classb*) and draw a line between them. The resulting *class diagram* will resemble a family tree. *HINT: Use the occurrences of the “new” command in the four Java files to find connections between the four classes.*

**Exercise 1C**: Compile and run the *Journey To Mars* program. Give the program a *whirl* to get a feel for what it does. While much of it might look like *Greek* to you now, try to relate some of the program’s functions to your analysis in the previous two exercises. *HINT: The right and left arrow keys make the worlds go ‘round.*

**Level 2**

Just as the clock strikes quittin’ time, you place the SDD on your boss’s desk and breathe a sigh of relief. With a quick glance, he reviews your work, and then looks up with a smile. “Nice job!” he says. “You have a good understanding of how this program works.” He points to a stack of papers on his desk and adds, “As you can see, I’ve got a bunch of anxious astrodynamicists on the third floor who won’t stop bugging me with change requests. To tell you the truth, I could give a rat’ aaa…” Pausing, he looks at the pile of papers, then turns back to you with a grin, “…aaactually, I am appointing you *team lead* on this project. I need the final version of the software delivered to the third floor engineering department no later than one week from today.” You stand dumbfounded in his office, and by the time you recover your senses enough to speak, your boss has left for the evening.

Over the next few days, you wait patiently in your cube for your *team* to arrive, but hear only the sounds of the ghostly toilets flushing. “Team lead, he said?” It’s quickly becoming clear that this is a team of *one*. If this project is going to get done, it’s all up to you! You glance at the pile of software change requests (now) on your desk and grab the one on the top. “I hope this is an easy one,” you mutter to yoursel

**Exercise 2A:** *.*

**Level 3**