For part2 of the lab we decided to use java. We chose java because of all of its useful libraries for writing objects to bytes and storing them to a file such as ByteOutputStream and ByteInputStream. These classes allowed us to easily write our superblock object to disk and read it from disk. Implementation wise we decided to represent the Superblock and Inodes as objects in java. The inode has the same fields as the initial assignment instructions. We use the serializable interface here in order to store and read the object using disk0. Inode also contains two functions. One to set the name of the inode which copies a character array and converts it to a string to store on disk. It also has a method to set up the block pointers which points to the file blocks. The superblock object has an overloaded constructor which takes in an array of bytes. This array of bytes would be the disk0 when a super block exists. When a superblock does not exist its constructor has no parameters and it initializes everything to default empty values. There are two methods which help with this readInodes which reads the inode data from disk to put into the array of inodes. Then there's a byteArraytoInt method which converts a byte array to an integer array. This method returns an array of size 1 for a single integer and an array of size n for n integers. The superblock object is also serializable allowing it to be written and read to and from disk.

FileSystem.java contains the main method. We decided to create a menu option to make the program more user friendly. The user can enter whichever operation they would like to do and then this will continue to loop allowing the user to do multiple additions with one run of the program. Choosing to exit will "save" the changes to the disk file. It is in this class we used the byteoutputstream class and the byteinputstream class. The byteoutputstream class in java writes an object to a byte array. The byteinputstream class reads bytes from a byte array and returns objects. This is what allowed us to get our superblock on and off disk.