1: Data stored in memory is different from file, memory data is stored directly in the computer's RAM and is easy to access by applications, but this makes it volatile, and the data will be lost when the program or computer are shutdown, file data is a long term solution to storing data, file data is stored in persistent storage devices like a hard drive or SSD and so can be accessed after a computer is shutdown and restarted. Files can also handle large amounts of data that RAM could not. Files need unique commands in order for their data to be accessed, used, and edited by Java programs.
2:import java.io.File;
3:The pathway given doesn't actually contain a path, just the name of the file, a file cannot be located with with path
4: The try/catch statement is used for exception handling
B: try {
textFile.createNewFile(insertpath);
System.out. println("File created");
} catch(IOException ie) {
System.out. println("File couldn't be created");
System.err.println("IOException: " + ie.getMessage());
}
5: System.err
B: There are a few locations, the message can be displayed in the console using System.err.println(); but you can also use JOptionPane to create pop up windows with the error message.
6: The file stream keeps track of where they are in the file, known as file position, file position is relative to what was last read/written and what will be read/written next.
B: The character Cr(carriage return) followed up by Lf(line feed) make a line terminator
7: The FileWriter and BufferedWriter classes are the classes used together to write data to a file.
8: while ((accountname = readFile.readLine()) != null) {
accountbalance = readFile.readLine();

Accountvalue = Double.parseDouble(accountbalance);
+ "\n");
totalbalance += Accountvalue;
}

9: Object serialization is when object data is written into a file for storage, and object deserialization is when object data is retrieved from a file for use.
10: The Serializable interface needs to be implemented in order to write objects to a file.