Conditional Statement in Java

**Programming example:**

* Steps:
  1. Describe
  2. inputs and outputs
  3. Problem analysis
  4. Algorithm design
  5. Variables
  6. Main algorithm
  7. Write code
  8. Test and debug

A real example:

* 1. **Describe**

**A program that writes to file and orders by**

* **Student ID**
* **Surname**

**Also can view the student details**

* 1. **Inputs and outputs**

**Inputs: the student data from the text, bubble sort methods,**

* **Student: surname, given names, preferred names, fulltime? Student ID, contact number, loving java? Continuing to diploma?**
* **User Input: Choose what to do**

**Outputs: 2 different text files ordered differently and console output**

* 1. **Problem analysis**
     + Read data from a text file
     + Bubble sort with two different methods
     + Write the data to 2 different text files in 2 different orders
     + Display online? Console?
  2. **Algorithm design**
     + Read from file
     + Give options to user
       - Bubble sort order by ID – write to file
       - Bubble sort order by surname – write to file
       - Display changes on console
       - End program
  3. **Variables**
* **Main Class**
  + - String[][] information;         //multidimensional Array, makes it easier to use bubble sort
    - bubbleSort method = new bubbleSort(); //initialize bubbleSort class
    - Scanner scan = new Scanner(System.in); //initialize scanner for user input
    - fileOptions read = new fileOptions(); //initialize fileOptions class
    - student[] studentChecker = new student[information.length]; //assigning returned values to newly made array, in order to choose which parts to access and how to display
    - Boolean invalid = true; // setting a while loop to keep program going until you manually leave
    - int numZeros;   //variable to count the number of zeros read
    - int numOdd;     //variable to count the number of odd numbers
    - int numEven;   //variable to count the number of even numbers
    - int choose // user input data, scans next int of what the user wants //else for error handling
* **Student Class // initialize them students, then the constructor, self-explanatory**
* **public** String surname;
* **public** String givenNames;
* **public** String preferredNames;
* **public** **int** studentId;
* **public** String fullTime;
* **public** **int** mobile;
* **public** String loveJava;
* **public** String diploma;
* **fileOptions Class**
  + - BufferedReader buff;    // reader
    - FileReader reader;   // simple reader
    - File filey;     // The file path
    - Writer writeOn;   // File writer
    - public String[][] readText() // reads text into the multidimensional array
    - values //the temporary string array list
    - temp // the other temp array list
    - returns // returns all the data
    - public void itWritesNow() // method to write to a file
    - jeff:studentChecker; // order and syntax of what to write to the file
    - writingToFile.close(); //close writing file method
* **bubbleSort Class**
  + - public student[] sortById(student list[], int listLength) //method to sort by ID
    - public student[] sortBySurname(student list[], int listLength) //method to sort by Surname
  1. **Main algorithm / Write the code**
* Firstly write attributing classes (bubbleSort, fileOptions and also create empty text files in resources AND link original text file as a resource.)
  + - Import scanner, IOexception
    - Initialize multidimensional array
    - Initialize other classes
    - Read from text file into a temp array, then into a multidimensional array where the second dimesion is a count function, to help me manipulate it after.
    - Repeat read until all students read in
    - Boolean to keep program looping with user options until manually ended
    - Give User input = choose scan.next int
      * Choose == 1 then bubblesort method write by ID (call from bubbleSort class)
      * Choose == 2 then bubblesort method write by surname (call from bubbleSort class)
      * Choose == 3 then show most recent bubbleSort to Console (method that recalls and shows)
      * Choose == 4 end program, breaks out of loop Boolean invalid = false
      * Else “invalid option, choose again” text
  1. **Test and Debug**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test description | Test steps | Data input | Expected outcome | Actual outcome |
| Run and choose 1 | **-run program**  **-choose 1** | **1** | **Writes to file by ID (check ID file to see)** | **Success (checked to see if It worked before)** |
| Run and choose 2 | **-run program**  **-choose 2** | **2** | **Writes to file by surname (check surname file to see)** | **As above** |
| Run and choose 3 | **-run program**  **-choose 3** | **3** | **Displays students on console by surname** | **Ditto, although “end of students msg displays for every student… fixed now** |
| Run and choose 4 | **-run program**  **-choose 4** | **4** | **Programs stops, can’t choose anything else** | **Gets out of loop after option 3… will look into. But end app works** |
| Run and choose 56 | **-run program**  **-choose 56** | **56** | **“invalid option, coose again”** | **Error handling works** |
| Run and choose a#tgoh | **-run program**  **-choose a#tgoh** | **a#tgoh** | **“invalid option, coose again”** | **Special characters break program ☹** |
|  |  |  |  |  |

**Does program do as client intended? Yes although the internet feature needs to be clarified… unsure what is intended to be done there.**