Department of Computer Science and Software Engineering
Xi'an Jiaotong-Liverpool University

Lab 4 – For Loops and Arrays

Aim

This lab aims to introduce you to for loops and arrays. You should make sure you have completed your week 3 lab, as we will use methods in this lab sheet. You will also use the scanner, so this will build on your previous weeks

Tips:

1. As stated last week, ask a TA if you have problems, they are here to help you!

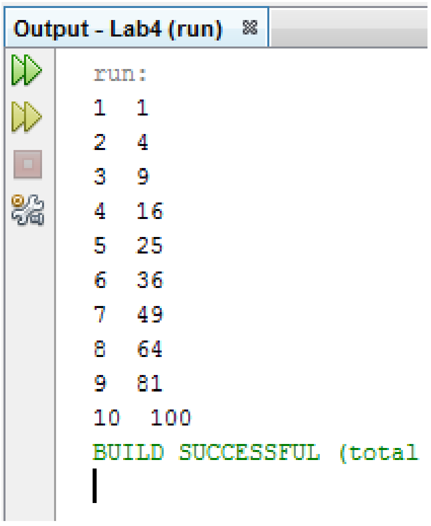
2. There is a checkpoint at the end of this sheet to evaluate your for loops. After this, the lab sheet will continue with arrays.

ShowSquares

The first task is to experiment with For loops, as covered in your week 4 lectures. First, create a class with a main method.

- In week 3 we introduced methods, so we expect that our main methods will now be short, and so you should create separate methods and call them.

- Here, we are going to create a method that will display the numbers 1 to 10, and their square number next to them, as shown below

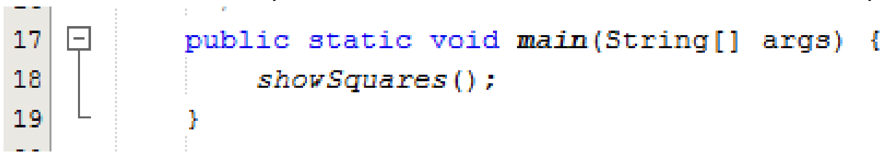


```
run:
1  1
2  4
3  9
4  16
5  25
6  36
7  49
8  64
9  81
10  100
BUILD SUCCESSFUL (total
```

- First, create a method called "showSquares()" that has no parameters, and is a void method (i.e. returns nothing).

- Use your week 4 lecture notes to create a loop from 1 to 10, and display the numbers. Then expand on it to also display the square of your number next to it, as shown above.

- Call this method from your main method. Your main method should only have one line of code!

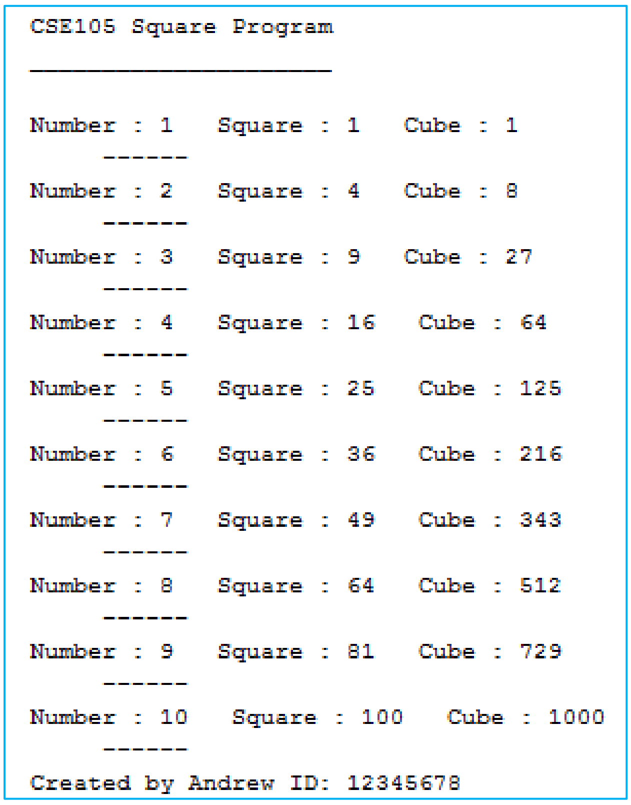


```java
    public static void main(String[] args) {
        showSquares();
    }
```

Extending ShowSquares

Once you have this method working, expend on it.

- Add a variable to calculate the cube of the numbers (i.e. 2 * 2* 2), and display it with the other numbers
- Make it more informative with additional information, such as getting it to say " Number : 1 Square : 1 Cube: 1"
- Add a decorative line (choose your own design!) after each line, so that it displays some decoration between each line
- Add a title of your own choice **before** the loop, and add your name and student number **after** the end of the loop
- The final result should look similar to the example below, but with your **OWN** design



```
CSE105 Square Program
_____________________

Number : 1   Square : 1   Cube : 1
     ------
Number : 2   Square : 4   Cube : 8
     ------
Number : 3   Square : 9   Cube : 27
     ------
Number : 4   Square : 16   Cube : 64
     ------
Number : 5   Square : 25   Cube : 125
     ------
Number : 6   Square : 36   Cube : 216
     ------
Number : 7   Square : 49   Cube : 343
     ------
Number : 8   Square : 64   Cube : 512
     ------
Number : 9   Square : 81   Cube : 729
     ------
Number : 10   Square : 100   Cube : 1000
     ------
Created by Andrew ID: 12345678
```

Editing your For Loop

The next step is to make some changes to your program to make it more interesting.

- Try initialising your loop with a value of 15. Change the condition to make sure it shows all values from 15 to 20:

```
CSE105 Square Program
_____________________

Number : 15   Square : 225   Cube : 3375
     ------
Number : 16   Square : 256   Cube : 4096
     ------
Number : 17   Square : 289   Cube : 4913
     ------
Number : 18   Square : 324   Cube : 5832
     ------
Number : 19   Square : 361   Cube : 6859
     ------
Number : 20   Square : 400   Cube : 8000
     ------
Created by Andrew ID: 12345678
```

- Can you make it count backwards from 20 to 15? You only need very small changes!

```
Number : 20   Square : 400   Cube : 8000
     ------
Number : 19   Square : 361   Cube : 6859
     ------
Number : 18   Square : 324   Cube : 5832
     ------
Number : 17   Square : 289   Cube : 4913
     ------
Number : 16   Square : 256   Cube : 4096
     ------
Number : 15   Square : 225   Cube : 3375
     ------
```

- Can you make it count backwards from 20 to 0, and show only every third value? You only need very small changes!

```
Number : 20   Square : 400   Cube : 8000
     ------
Number : 17   Square : 289   Cube : 4913
     ------
Number : 14   Square : 196   Cube : 2744
     ------
Number : 11   Square : 121   Cube : 1331
```

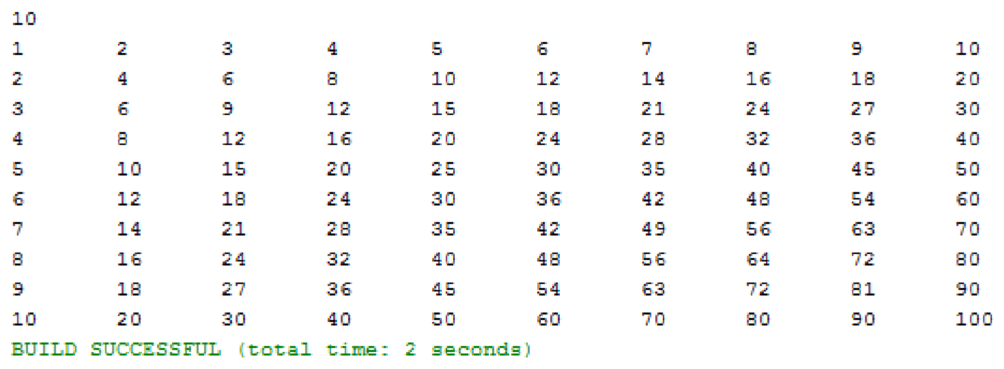
```
     ------
Number : 8   Square : 64   Cube : 512
     ------
Number : 5   Square : 25   Cube : 125
     ------
Number : 2   Square : 4   Cube : 8
     ------
```

- Finally, earlier in the semester, we used the Mod (%) value. Adjust the for loop above to add an if statement that only displays if the value can be divided by 2

```
Number : 20   Square : 400   Cube : 8000
   ------
Number : 14   Square : 196   Cube : 2744
   ------
Number : 8   Square : 64   Cube : 512
   ------
Number : 2   Square : 4   Cube : 8
   ------
```

Nested For Loops

The lecture introduced nested For loops. These are loops within loops. Here, you will use the nested loops to display a multiplication table, as shown below



```
10
1       2       3       4       5       6       7       8       9       10
2       4       6       8       10      12      14      16      18      20
3       6       9       12      15      18      21      24      27      30
4       8       12      16      20      24      28      32      36      40
5       10      15      20      25      30      35      40      45      50
6       12      18      24      30      36      42      48      54      60
7       14      21      28      35      42      49      56      63      70
8       16      24      32      40      48      56      64      72      80
9       18      27      36      45      54      63      72      81      90
10      20      30      40      50      60      70      80      90      100
BUILD SUCCESSFUL (total time: 2 seconds)
```

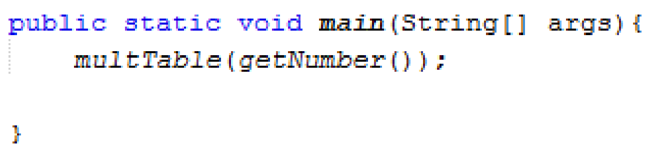
- Create a new void method with no parameters that will display your multiplication table. You will need to experiment with for loops

- HINT, use the tab character \t in your System.outs to align your text

- HINT 2, built it up slowly. Try one for loop, test it, then move on to the next loop

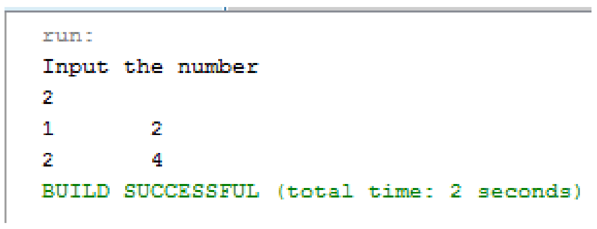
- Use the Scanner to input an integer to decide how large the times table should be, and input this as an argument to your method. You will need to change the method parameters

- The scanner input should be in a separate method! Your main method should only have one line of code

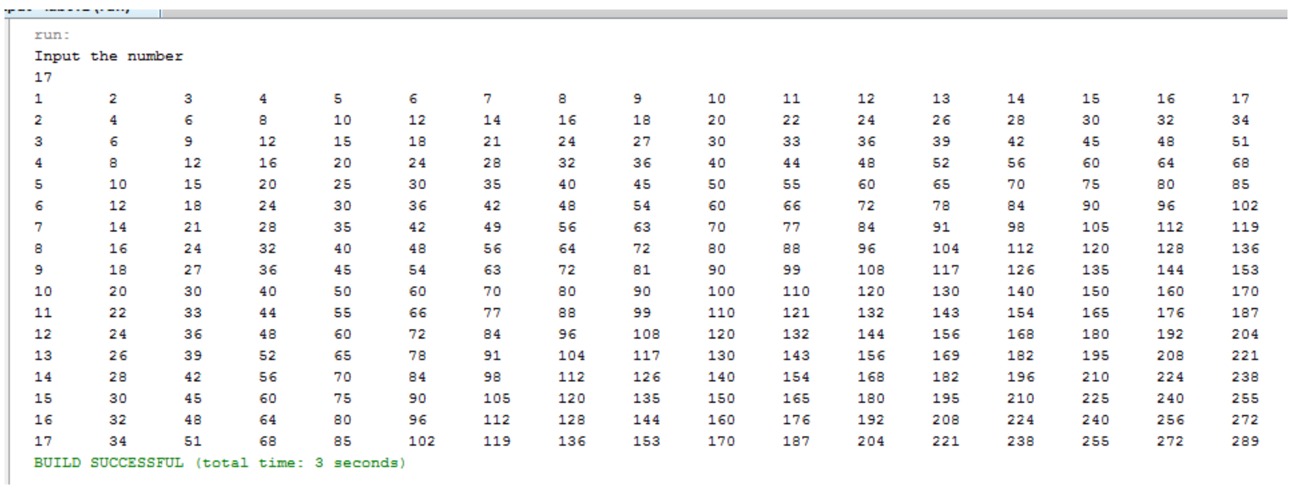


```
public static void main(String[] args){
    multTable(getNumber());

}
```



```
run:
Input the number
2
1       2
2       4
BUILD SUCCESSFUL (total time: 2 seconds)
```



```
run:
Input the number
17
1       2       3       4       5       6       7       8       9       10      11      12      13      14      15      16      17
2       4       6       8       10      12      14      16      18      20      22      24      26      28      30      32      34
3       6       9       12      15      18      21      24      27      30      33      36      39      42      45      48      51
4       8       12      16      20      24      28      32      36      40      44      48      52      56      60      64      68
5       10      15      20      25      30      35      40      45      50      55      60      65      70      75      80      85
6       12      18      24      30      36      42      48      54      60      66      72      78      84      90      96      102
7       14      21      28      35      42      49      56      63      70      77      84      91      98      105     112     119
8       16      24      32      40      48      56      64      72      80      88      96      104     112     120     128     136
9       18      27      36      45      54      63      72      81      90      99      108     117     126     135     144     153
10      20      30      40      50      60      70      80      90      100     110     120     130     140     150     160     170
11      22      33      44      55      66      77      88      99      110     121     132     143     154     165     176     187
12      24      36      48      60      72      84      96      108     120     132     144     156     168     180     192     204
13      26      39      52      65      78      91      104     117     130     143     156     169     182     195     208     221
14      28      42      56      70      84      98      112     126     140     154     168     182     196     210     224     238
15      30      45      60      75      90      105     120     135     150     165     180     195     210     225     240     255
16      32      48      64      80      96      112     128     144     160     176     192     208     224     240     256     272
17      34      51      68      85      102     119     136     153     170     187     204     221     238     255     272     289
BUILD SUCCESSFUL (total time: 3 seconds)
```

Checkpoint

Show a TA your nested loops times table method with a scanner method to choose how large it should be

Answer any questions they have, and then continue with the sheet

This will count towards your final grade

Arrays

So far, you have seen variable types such as String, int, and double. However, it can be useful to store a group of values together in arrays, as discussed in lecture week 4.

- Create a new class called ArrayWork. Give it a main method, and configure your project so that this is the main class. Consult your previous practicals if unsure what this means.

- An array needs to be declared, and space allocated. Following the lecture notes, create an array of 5 integers:
  `int[] intArray = new int[5];`

- We now want to initialise the array. Remember from the lecture that the first element of an array is the 0th element. We can therefore initialise the first value as `intArray[0] = 27;`

- Initialise all 5 elements. Then, following the lecture notes, use a System.out to display them. You have made your first array!

String Arrays

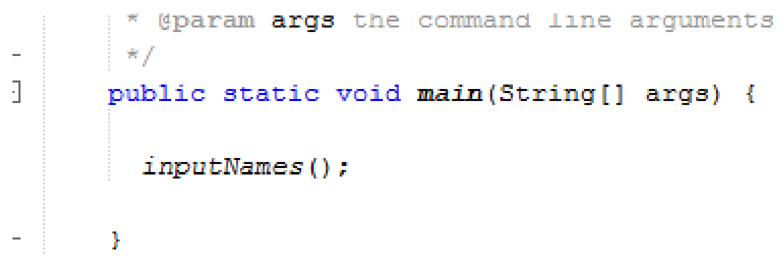
- Create a new Class called PersonDetails, and in the main method, use your lecture notes to create a new String Array of 5 Strings. Initialise it with 5 names

- Use a **For** loop to display all the names with a System.out.

- This is very basic, and we want to improve it. Create a void method called inputNames. Here, declare an array of 3 names, and then use a scanner input and a for loop to fill the array with Strings.

- At the end of the method, you should then pass the array into a "displayNames" method, which will display the output in a slightly nicer format

- At the end of this, you should have 3 methods. A main method, a displayNames method, and an inputNames method. Your main method should only have one line of code!

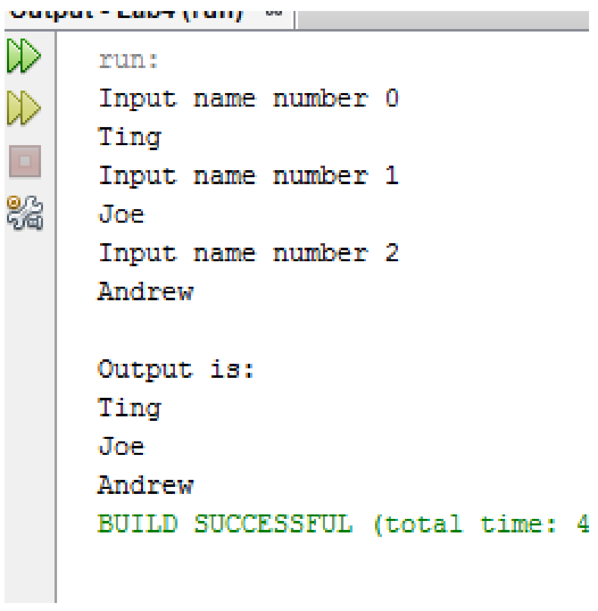


```
     * @param args the command line arguments
     */
    public static void main(String[] args) {

      inputNames();

    }
```

- Input should be handled as an array in that method, and then passed to the display method for output



```
run:
Input name number 0
Ting
Input name number 1
Joe
Input name number 2
Andrew

Output is:
Ting
Joe
Andrew
BUILD SUCCESSFUL (total time: 4
```

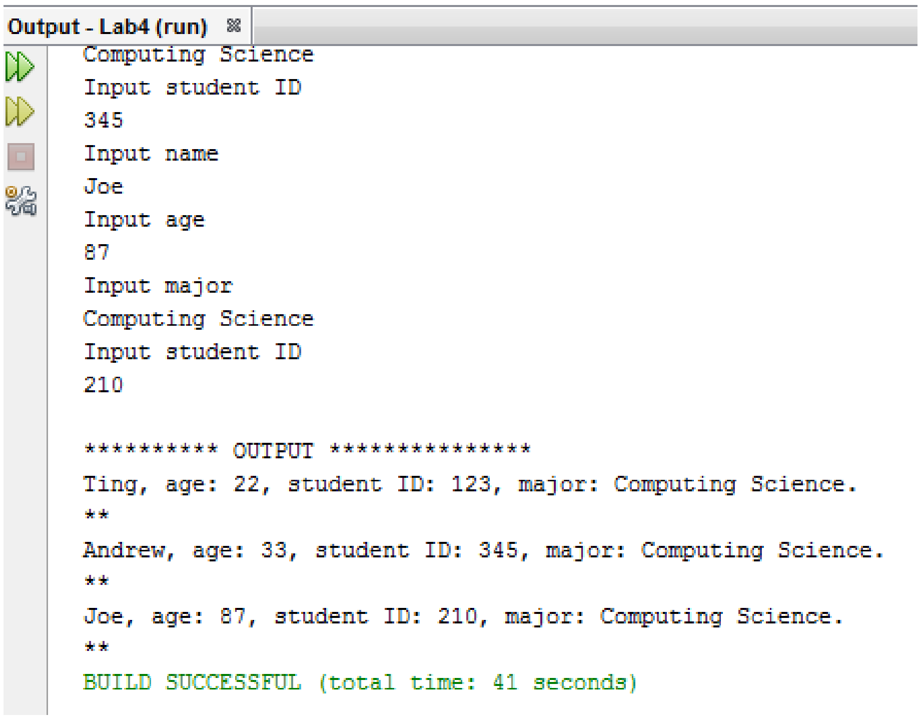
More Arrays

We will now extend our methods to handle information needed to store details about a person. A person has a name, an age, a major, and a student ID number. Here, we will create arrays to store this information, fill them, and then display information

- Declare new int arrays for age and student ID, and a new String array for student major

- Extend your input to receive keyboard entry for all of these. Use the nextLine method, and conversion methods to convert from a String to integer (see your lecture notes)

- Finally, pass all of these into an improved Display method. This will apply some formatting, and display all information. The complete program input and output should look like the example below. Remember, your main method should still only have **1 line of code**.



```
Computing Science
Input student ID
345
Input name
Joe
Input age
87
Input major
Computing Science
Input student ID
210

********** OUTPUT ***************
Ting, age: 22, student ID: 123, major: Computing Science.
**
Andrew, age: 33, student ID: 345, major: Computing Science.
**
Joe, age: 87, student ID: 210, major: Computing Science.
**
BUILD SUCCESSFUL (total time: 41 seconds)
```

More Arrays

We can also use our arrays to calculate useful information. The lecture notes discuss how to calculate information with an array. We created a number of arrays, including an array of ages.

- Create a method that will receive an array of ages, and will calculate the average age.

- Use your lecture notes as guidance, and think about how you would calculate an average. Ask a TA and work with your classmates!

- Display the sum total and the average age in the method using system.outs.

Next Steps – Complete Previous Homework

When you have completed this sheet, try to complete all your homework sheets from previous weeks. By this point, you should have finished all of the sheets!

Created Autumn 2017, Modified Autumn 2018
By Dr Andrew Abel