

Lab 2

For this lab you will write a program that prompts the user for a decimal (base 10) integer and also for a base between 2 and 36, then perform a base conversion of the integer to the new base and print the result to the console. **Don't be afraid to ask for help. The purpose of this lab is to continue to refresh your Java skills.**

1. Using IntelliJ, create a project called `Lab2`. Add a class called `BaseConversion` with a `main` method. If you are working on an Otterbein machine, you should either create your project somewhere other than the default location on the C: drive (like `OneDrive`) or make a copy of it, since the local drives on these machines are wiped regularly.
2. The input to this program will be two integers. You should read these values into `int` variables with a `Scanner` object that you create at the beginning of the program. Use its `nextInt()` method to read these values. If you don't remember how to create a `Scanner` object, consult the *Start Concurrent* textbook.
3. Error handling: Although you can assume that the user enters correctly formatted integers for both the decimal integer and the base, you may not assume that the user will enter a valid base. Legal bases are between 2 and 36, inclusive. If the user enters a base outside of this range, print `Illegal base!` and do not attempt to perform the base conversion.
4. There are a few algorithms for converting a number `n` represented in base 10 to base `b`. The algorithm we will use has two parts. First, we find the largest power of `b` that is no greater than `n`. To do so, I recommend repeatedly multiplying a variable representing this power by `b` as long as it is less than or equal to `n/b`. (Alternatively, you can multiply this variable by `b` until it is greater than `n`, but doing so will require you to divide by `b` afterwards. Let's call this value `power`, which is a power of `b` no larger than `n`.

Repeatedly, as long as `power` is greater than zero, divide `n` by `power`. Doing so will produce a representation of your digit. If the digit is less than 10, you can simply print it out. If it's greater than or equal to 10, you will need to print out the character 'A' through 'Z' that corresponds to the digit value. To do so, subtract 10 from the value, add 'A', and cast the result to a `char` before printing it out.

Once you've determined the current digit, set `n` equal to the result of itself modulus `power`, which will give you the value of `n` that is below the current place value of `power`. Finally, divide `power` by `b` so that you're looking at the next lower power of `b`.

Eventually, power will become 0, and your task will be done. Note that it is difficult to determine how many digits a value will have in a given base ahead of time. Thus, the loop that allows you to repeat an unknown number of times is a good fit. Indeed, you will probably use this kind of loop twice: first to determine the value of power and second to extract each base b digit.

5. Below are three examples of sample output. Your program should match these examples as closely as possible. Testing with other inputs is also a good idea.

The first example converts 27 to base 2.

```
Enter a number: 27
Enter the base you want to convert to: 2
27 in base 2 is: 11011
```

The second example converts 115 to base 13. For bases higher than 10, there's a chance that the letters A-Z will be needed to represent digits with a value of 10 or greater. In this case, the digit B represents 11.

```
Enter a number: 115
Enter the base you want to convert to: 13
115 in base 13 is: 8B
```

The third example shows an error case in which the user asked for an illegal base. Only bases 2 through 36 are supported.

```
Enter a number: 342
Enter the base you want to convert to: 40
Illegal base!
```

6. **Deliverable:** Find the file `BaseConversion.java` in the `Lab2\src` folder inside your workspace folder. Unless you changed it, your workspace folder should be `C:\Users\<<your_user_id>\IdeaProjects\`. If you are working on an Otterbein machine, you should either create your project somewhere else (like OneDrive) or make a copy of it, since the local drives on these machines are wiped regularly. Email a zipped copy `BaseConversion.java` to prof. stucki. Do not email the entire project. I only want the `BaseConversion.java` file.

All work must be done individually. Never look at someone else's code. Please refer to the course policies if you have any questions about academic integrity. If you have trouble with the assignment, I am always available for assistance.