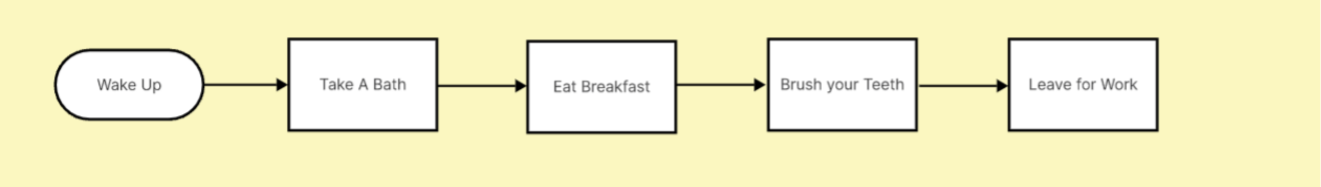
Intended Learning Outcomes

1. Define Algorithms and its Applications
2. Ways on presenting an Algorithms

What are Algorithms?



- When you say Algorithms. These are basically ***procedures*** or ***steps*** to solve a particular problem ***systematically***. It could be in relation to any problem because the word Algorithm is ***not only exclusive*** to the world of programming.
- Algorithms are found ***everywhere***! In fact, one of the most common examples of an algorithm that you do (that you probably do not notice) is your morning routine.
- ***Consider the image above:*** Notice how most of the time that after you wake up, you always do this routine or preparation before going to work, school, or to start the day in general. Algorithms are not hard to define, its simplicity can be understood by everyone. However, despite it being simple, we can integrate it with complex concepts just like in the field of Electronics, how Alternating current is transformed into Direct current. In the field of networking, how data travels using the OSI Model. In the field of Mathematics, where we apply different theorem to a Differential Equation to transform the General Solution to a Particular Solution and so much more.

Presenting and Reading Algorithms

So how are we suppose to read an algorithm?

- Well, we must first visualize and create a plan to attack the problem. We could make different models, sketches, or drawings that would help visualize our plan but there are other methods that are solely based on "formality" or what is "common" for most people who are also involved in this discipline (which will be discussed in this module).
- ***So what are these ways to visualize an algorithm?*** One formal way on presenting your plan of attack to a problem is by the use of ***Flow Charts***. Im sure that you are familiar with Flow charts but if not, one example is the image that I used above. As you can see, the flow chart is composed of many components and these components include the following:
- ***Shapes*** - Which represent a step or procedure (Rectangle for Process, Diamond for Condition, etc)
- ***Arrows*** - Which references to the next step.

Another way of presenting algorithms is by the use of Pseudocode

- Pseudocode, is another way to present an algorithm, but instead of using shapes and lines, it uses some ***terminologies that you often find in programming languages***.
- Moreover, note that when using this method of presenting an Algorithm. Your ***pseudocode does not need to be perfectly inline with the programming language that you chose***.

Consider the code below:

```
'''
list of numbers

for number in list of numbers
  if number in list of numbers
    print True
  else
    print False

'''
```

- In the example code above, I used Python as a basis to create a pseudocode. A pseudocode is a code that is not entirely correct with regards to how a conditional statement is written in Python. However, it is somewhat identical to how a Python conditional statment is written.
- The key here is to ***show how the algorithm works***. In the example code above, notice how it is about finding a certain number in the list of numbers. The numbers prints out a string "True" if the number is inside the list or else it prints out "False".

Conclusion

Some key takeaways for this module:

1. *Algorithms are basically steps and procedures to solve a particular problem*
2. *Flow charts, and Psuedocode are the some of the formal ways to present an algorithm in the programming world.*
3. *Algorithms can be found everywhere, wether it be the simplest or the most complex concepts.*

Next Topic

2. ***The Two Measures for Algorithms***