



## **Task: Control Structures I**

[www.hyperiondev.com](http://www.hyperiondev.com)



# Introduction

## Welcome to the Control Structure Task!

Please feel free to visit [www.hyperiondev.com](http://www.hyperiondev.com) and view the tools and methods that will help you throughout the course.

For any queries regarding the course, need help understanding the task or general comments, please contact us at [help@hyperiondev.com](mailto:help@hyperiondev.com).

---

In this task you will learn about a program's flow control. A control structure is a block of programming that analyses variables and chooses a direction in which to go based on given parameters, in essence, it is a decision-making process in computing that determines how a computer responds when given certain conditions and/or parameters.



*Let's consider how we make decisions in real life? First you are faced with a problem, then you have to see what the problem entails. Once you figure out the crux of the problem you then follow through with a way to solve this problem based on what you were presented with. Teaching a computer how to solve problems works in a similar way. We tell the computer to look out for a certain problem and tell it how to solve the problem when faced with it.*

## What is Computational Thinking?

*Computational Thinking is the process of solving a problem using a certain manner of thinking. Computational Thinking is used in the development of computer applications but it can also be used to solve problems across a wide variety of categories, including maths, science, geography and art. After learning about computational thinking you will be able to apply this to your school work as well as real life problems outside of school.*

## Simple Daily Examples of Computational Thinking

- Looking up a name in your contact list
  - Standing in lines at the bank, mall or home affairs
  - Packing your bag for school
  - Cooking a gourmet meal
  - **Hyperion Team**
-

## If Statements:

We are now going to learn about a vital concept when it comes to programming. We will be teaching the computer how to make decisions for itself. This concept is called an **If Statement**. As the name suggests it is essentially a question. It is able to compare two or more variables or scenarios and perform a certain action based on the outcome of that comparison.

An example of an if statement in our everyday life would be: If it's raining outside then you should take an umbrella with you. So before taking an umbrella with you, you would first go outside and see if it's raining before making that decision.

In programming an if statement would look like this:

```
num = 10
```

```
If (num < 12):
```

```
    Print "the variable num is lower than 12"
```

So the statement checks if the variable number is less than 12. If it is, then it will print the sentence letting us know. If it num was greater than 12 then it would not print out that sentence.



A note from Masood...

*Sorry to interrupt but I found this fantastic blog WRITTEN by [MARC CHERNOFF](#) that showed me 10 If Statements in "Real Life" that are worth learning. It just shows you how important making decisions are to our lives. It also shows you that if you don't follow the desired path based on your decision the outcome could be detrimental to you .*

- 1. If you don't understand the product or service, don't buy it until you do.*
- 2. If you do not take ownership of your actions, your actions will eventually own you.*
- 3. If you are not saving at least 10% of your salary, you are not saving enough.*
- 4. If you talk too much, people will stop listening. If you don't talk enough, people will never hear your point of view.*
- 5. If you are lazy, you will fail. Laziness will always overshadow your true potential.*

6. *If you hate your job, you also hate half of the time you spend on this planet.*
7. *If you are not investing (120 minus your age) percent of your savings in the stock market, you are giving up thousands of dollars over the course of your lifetime.*
8. *If you don't finish what you start, your success rate will always be zero.*
9. *If you don't consume enough liquids, you will never be healthy.*
10. *If your monthly debt payments exceed 40% of your total income, you will go broke if you don't fix your spending habits promptly.*

## - Masood Gool

---

As you can see the **if statement** is pretty limited as is. You can only really make decisions based if there are two outcomes. What happens if we have more options? What if based on one decision it will have further ramifications and we will need to make more decisions to fully solve the problem at hand?

Control structures are not limited to **If statements** as you will find out in the next 2 tasks. You will learn on how to expand on an If statement by adding an **Else statement** or an **Elif statement** (Else If).

---

## Instructions

First read example.py, open it using Notepad++ (Right click the file and select 'Edit with Notepad++').

- Example.py should help you understand some simple Python. Every task will have example code to help you get started. Make sure you read all of example.py and try your best to understand.
- You may run example.py to see the output. The instructions on how to do this are inside the file. Feel free to write and run your own example code before doing the Task to become more comfortable with Python.
- You are not required to read the entirety of Additional Reading.pdf, it is purely for extra reference.

# Compulsory Task

## Follow these steps:

- Create a Python file called “baby.py” in this folder.
- This program will be used to test if the user is over 18 and allowed entry into a party.
- Ask the user to enter the year they were born and calculate if they are older than 18.
- If they are older then display a message saying “Congrats you are old enough”

## Things to look out for:

1. Make sure that you have installed and setup all programs correctly. You have setup **Dropbox** correctly if you are reading this, but **Python or Notepad++** may not be installed correctly.
2. If you are not using Windows, please ask your tutor for alternative instructions.

## Still need help?

There are different ways to get assistance from your mentor and the Hyperion Team

- Login at <https://www.hyperiondev.com/support> to see all the ways you can get support - including chatting to your mentor.
- Write your queries in your comments.txt file and your mentor will respond
- Visit [Hyperion Help](#) for frequently asked questions
- Lastly you can email us on [help@hyperiondev.com](mailto:help@hyperiondev.com).

# Task Statistics

Last update to task: 29/09/2016

Author: Masood Gool

Main mentor: Umar Randeree

Task Feedback link: [Hyperion Development Feedback](#).