

Introduction to Programming in Turing

Input & Variables

The IPO Model

The most basic model for a computer system is the Input-Processing-Output (IPO) Model.

In order to interact with the computer as a programmer, we must develop simple examples of each of these stages, which we will then build upon to solve more and more sophisticated problems.

Storage & Variables

In order to interact with the outside world, programs have to accept Input. Before doing this, however, we need to discuss storage and variables.

If the computer asks the user for their name, age, grade, address, or any other information, it should remember that information.

We store information in memory, and the specific location in memory is called a variable (because its value varies).

Variables & Data Types

When we declare a variable:

1. a space is reserved in memory for that data
2. a name is reserved to identify that data

For now, we will consider only the three most basic types of data. When we declare variables, we also specify the data type. This is done to help the computer understand what we expect to use the variable for.

Data Types

string – a string value is a collection of characters, such as a name, address, or other combinations of letters and numbers

int – an integer value is a positive or negative whole number (... , -3, -2, -1, 0, 1, 2, 3, ...)

real – a real number involves decimals, such as 0.5, 0.33, 10.7. You can also represent integers as reals, but try to avoid this (-3.0, 4.0)

Declaring Variables for: strings, integers, real numbers

```
var firstName : string
```

```
var age : int
```

```
var bankBalance : real
```

The keyword “var” is used to declare a variable. Then we give a meaningful name, and after the colon (:), identify the type of variable (string, int, real)

A Sample Program

For input, use the “get” statement

```
var firstName : string
```

```
put "Please enter your name"  
get firstName
```

```
put "Hello "  
put firstName
```

Improving the Interface

Quite often, we want to combine our input and output on the same line. In Turing, use the two periods (..) at the end of the “put” line.

```
var firstName : string
```

```
put "Please enter your name: " ..  
get firstName
```

```
put "Hello " ..  
put firstName
```


Input Entire Line to Variable

By default, the get command will break up input wherever there is a space. Sometimes we want the entire line, with spaces, in a single variable (e.g., an address).

```
var address : string
```

```
put "Please enter your address: " ..  
get address : *
```

The “ : * ” after the get command instructs Turing to take the whole line, with spaces.