Introduction to Programming in Turing

Input, Output, and 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.

Output

The most fundamental operation of a computer program is to communicate something useful to the end user.

We accomplish this with the concept of Output. Output is implemented in many ways depending upon the programming language.

For example: put, print, printf, puts

Using the “put” Command

In the Turing programming language, output is accomplished using the “put” command.

To start, we will consider outputting the most basic types of data – strings (which are groups of characters), and integers (whole number values).

Output in Turing

Whenever we refer to a string in Turing (and most other languages, and pseudocode), we need to put the characters in quotation marks:

```
put “Hello!”
put “Hello world!”
```

For integers, no quotation marks are needed. Turing recognizes that they are numbers.

```
put 35
```

Mathematical Operations

To actually be making use of the computer's calculating ability, we need to use some mathematical operators.

Operator	Operation	Code
+	Add	A + B
-	Subtract	A - B
*	Multiply	A * B
/	Divide	A / B
**	Exponent	A ** B

Math using Turing

put 3 + 5	output is 8
put 4 – 11	output is -7
put 2 * 6	output is 12
put 7 / 2	output is 3.5

Remember that Order of Operations (BEDMAS) applies to what you are doing. You can use brackets to ensure calculations are done in the order you want.

$$\frac{3+4}{2-5}+\frac{11\times 6}{2^3}$$

(3 + 4) / (2 – 5) + (11 * 6) / (2**3)

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
```

Exercises - setup

You should have a folder for this unit:
H:\00 ICS3M\02 Turing Basics\

Create a sub-folder called “Exercises”

Save each exercise as you work through it, using consistent naming

(exercise01a.t, exercise01b.t, etc...)

Exercises

1. Write a program that prompts the user for their name, age, and address, and then outputs this information using one or two nicely formatted sentences. (exercise01a.t)

2. Write a program that asks the user to input marks for 4 courses, and the displays the average mark. (exercise 01b.t)

3. Ask 5 simple math problems. After the user enters their answer, display the problem and the correct answer. The correct answer should be calculated by your program. (exercise01c.t)

Exercises…

4. Ask the user their name, age, and the current year. Greet the user and tell them the year they will be 25, 50, and 75. (exercise01d.t)

5. Have the user enter two integers between 1 and 9. Do the basic math operations using these numbers and display the answer (add, subtract, multiply, divide, exponent). (exercise 01e.t)

6. Using the available help guide for Turing, investigate how to do #2 with your answer rounded to one decimal place using “put”. (exercise 01f.t)