Software Development 1
ASCII, Inputs

TUDublin- Tallaght Campus

E.Costelloe

Inputting Values into Variables

- Up until now we have been completing calculations using variables, that are essentially hard coded.

- These program's would not be very productive, if they always calculated the same mathematical equation, or processed the same data.

- Most programs require user input to complete their processing. For now we will look at inputs from the keyboard.

- But this is just the beginning, later you will use data from a multitude of other inputs: text files, csv files etc. …

Inputting Values

- Usually in most programming languages, they use an I/O stream to obtain data or output data.

- A stream can represent many different kinds of sources and destinations, including disk files, devices, other programs, and memory arrays. Streams support many different kinds of data, including simple bytes, primitive data types, and objects. Some streams simply pass on data; others manipulate and transform the data in useful ways.

- No matter how they work internally, all streams present the same simple model to programs that use them: A stream is a sequence of data. A program uses an *input stream* to read data from a source, one item at a time:

Inputting Variables

- Python has (for now) two I/O streams. Input and Output
  - *Output you have already seen => print("hello")*
  - *Input – we will look at next.*

  - Python and other languages only allow you to input a string.

  - But this in itself causes an issue?

    "If I read in a number as a string, how can I multiply it for example by itself?"

  - Any suggestions on a solution?

    - Previously we looked at casting (converting from one data type to another).

Inputting Variables

- Lets look at inputting a string first as it is the default:
- The function for taking in a string from the keyboard is simply “input()”.

It is the simplest way to read user input, for instance, input(): it prints its optional argument string as a prompt and returns the user’s typed reply as a **string**.

- Optionally accepts a string that will be printed as a prompt (e.g., input('Press Enter to exit'))
- Returns to your script a line of text read as a string (e.g., next_input = input())

Example : to read your name from the keyboard:

```
name = input()
print("Hello", name)
```

- There is a bit of an issue with this however.
- You have not told the user what to do…. The screen is blank.

Inputting Variables

- Generally you must prompt the user what to enter:

- You can add a prompt:

```
print("Please enter your name:", end="")
name = input()
print("Hello", name)
```

- The end=“” is to stop the cursor going to the next line (cosmetic)

Inputting Variables

- Generally you must prompt the user what to enter:

- Python allows you to add the prompt to the input:

```
name = input("Please enter your name:")
print("Hello", name)
```

- This (as a concept) adds in the ,end=””
- What if you wanted the opposite, the input on the next line?

Inputting Variables

- Generally you must prompt the user what to enter:

- Input on next line:

```
name = input("Please enter your name:\n")
print("Hello", name)
```

- This takes in a string, what about a integer or a float?

Inputting Variables

- This takes in a string, what about a integer or a float? Use casting to convert the values from strings to the appropriate data type

- 1) →

```
age = int(input("Please enter your age:"))
```

or

- 2) →

```
age = input("Please enter your age:")
age = int(age)

print("You are", age, "years old!")
```

Method 1 and 2 produce the same result.

Inputting Variables

- This could be the first time that you get a runtime error.

```
age = int(input("Please enter your age:"))
```

```
age = input("Please enter your age:")
age = int(age)
```

-

```
print("You are", age, "years old!")
```

- A **runtime error** is produced in this case by…….entering text where numeric data should be input e.g. entering “sdfs” will generate the following error

```
ValueError: invalid literal for int() with base 10: 'sdfs'
```

Inputting Variables

- Try in class:

- Write a program that asks the user for the price of a product and the tax rate for that product. Your program should calculate and output the final cost of the product after tax being added and outputs the tax total.

- Pseudocode first

  Data

  Algorithm Steps

Inputting Variables

Inputs:

price : FLOAT

tax : FLOAT

Outputs

tax_amount : FLOAT

final_cost :FLOAT

Begin

1. *Output(Title)*
2. *Output("Enter initial cost of product ")*
3. *Input(price)*
4. *Output("Enter tax rate for product ,e.g. 10.00 for 10% ")*
5. *Input(tax)*
6. *tax_amount = price * (tax / 100);*
7. *final_cost = price + tax_amount*
8. *Output(" Cost before tax: ")*
9. *Output(price)*
10. *Output ("Final Cost")*
11. *Output (final_cost)*
12. *Output("Total tax ")*
13. *Output (tax_amount)*

End

Inputting Variables

- Try in class:

- Edit the previous program to take in the name of the product from the user and use it in the output display

- :
- :