

# C0452

# Programming Concepts

Lecture 2

How to analyse User Requirements

**“Where do I start?”**

# My problem-solving process

- **Identify concepts** in sentences (one line at time)
- Identify the **core functionality** (summarise the aim)
- **Implement** the concepts identified

# Rainfall problem

Problem: Read in integers that represent daily rainfall, and print out the average daily rainfall. If the input value of rainfall is less than zero, prompt the user for a new rainfall. When you read in 99999, print out the average of the positive integers that were input other than 99999.

Identify concepts

# Line by line

Verb (doing word): method

data type: int

Problem: **Read in integers** that represent daily rainfall,

# Line by line

Verb (doing word): method

data type: int

Problem: **Read in integers** that represent daily rainfall, and **print out the average** daily rainfall.

Verb (doing word): method

calculation: total / number

# Line by line

Verb (doing word): method

data type: int

Problem: **Read in integers** that represent daily rainfall, and **print out the average** daily rainfall.

Conditional!

Verb (doing word): method

calculation

The condition:  $\text{input} < 0$

**If the input** value of rainfall is **less than zero**, **prompt the user for a new** rainfall.

Output a message...

... asking for a new input



# Line by line

Verb (doing word): method

data type: int

Problem: **Read in integers** that represent daily rainfall, and **print out the average** daily rainfall.

Conditional!

Verb (doing word): method

calculation

The condition:  $\text{input} < 0$

If the **input** value of rainfall is **less than zero**, **prompt the user for a new rainfall**.

Output a message...

... asking for a new input

Conditionall if statement

Conditional!

The condition:  $\text{input} == 99999$

Verb (doing word): method (see above)

**When you read in 99999, print out the average of the positive integers that were input other than 99999.**

Assuming correct implementation of previous sentence, there would be no negative values stored?

Identify the problem

## So what's the problem here?

Problem: **Read in integers** that represent daily rainfall, and **print out the average** daily rainfall.

**If the input** value of rainfall is **less than zero**, **prompt the user for a new** rainfall.

**When you read in 99999**, **print out the average** of the **positive integers that were input** other than 99999.

# Which one of the sentences?

Problem: Read in integers that represent daily rainfall, and print out the average daily rainfall.

**If the input value of rainfall is less than zero, prompt the user for a new rainfall.**

**When you read in 99999, print out the average of the positive integers that were input other than 99999.**

# Why?

Problem: Read in integers that represent daily rainfall, and print out the average daily rainfall.

**CONDITIONAL:** **If** the input value of rainfall is less than zero, prompt the user for a new rainfall.

**CONDITIONAL:** **When** you read in 99999, print out the average of the positive integers that were input other than 99999.

Implement core  
functionality first

# Build the main functionality

Verb (doing word): method

data type: int

- 1) **Read in integers** that represent daily rainfall
- 2) **Print out the average** daily rainfall

Verb (doing word): method

calculation: total / number

# Then add conditionality

Verb (doing word): method

data type: int

- 1) **Read in integers** that represent daily rainfall
- 2) **Print out the average** daily rainfall

Verb (doing word): method

calculation: total / number

- 3) **If input is less than zero...**
- 4) **When input is 99999...**

These conditions can be added to compliment the main functionality  
– where are these conditional statements best placed?



Identifying entities

# Example description:

“This app is a simple simulation of a ticket machine where a user can enter real money as coins and is able to purchase tickets. Edit your copy of the project to offer tickets to three local stations:”

1. Aylesbury costing £2.20
2. Amersham costing £3.00
3. High Wycombe costing £3.30

# Find the 'entities' (groups of 'things')

“This app is a simple simulation of a **ticket machine** where as user enters real money as **coins** and is able to purchase **tickets**. Edit your copy of the project to offer tickets to three local stations:

1. Aylesbury costing £2.20
2. Amersham costing £3.00
3. High Wycombe costing £3.30”

# Represent each entity as a class!

Ticket

TicketMachine

Coins

# Analysing user requirements

# Example requirements

1. Each ticket should have a destination, a cost, and the date purchased
2. Print a list of all the available Tickets
3. Select one of the Tickets to purchase

# Analysing these requirements

each ticket object

Variables – attributes of each the ticket object

1. **Each ticket** should have a **destination**, a **cost**, and the **date purchased**

# Analysing these requirements

each ticket object

Variables – attributes of each the ticket object

1. **Each ticket** should have a **destination**, a **cost**, and the **date purchased**

Verb (doing word): method

ticket objects

2. **Print** a list of **all** the available **tickets**



# Analysing these requirements

each ticket object

Variables – attributes of each the ticket object

1. **Each ticket** should have a **destination**, a **cost**, and the **date purchased**

Verb (doing word): method

ticket objects

2. **Print** a list of **all** the available **tickets**

Conditional?

One ticket object

3. **Select one** of the **tickets** to purchase