# the Master Course

{CODENATION}

# Brown Bag

## Problem Solving

{CODENATION}

# The Importance of Problem Solving

It not only applies to software development but
to **all careers** and daily life

{CODENATION}

# The most important skill

> Code is just **a tool to solve a problem**
> Software development is as much about the tools as it is the **mindset**

{CODENATION}

# What is Problem Solving?

The process of **finding solutions** to difficult or complex issues.

{CODENATION}

# How to problem solve...

...Lets break it down

{CODENATION}

**Problem** - A state of desire for reaching a definite goal from present condition

**Solution** - The management of a problem in a way that successfully meets the goals set for treating it

**Have a goal!**

# Identify the Problem...

...Defining and verbalising the problem
**establishes a goal**

**Input** - something is wrong or something could be improved

**Output** - a clear definition of the opportunity and a goal for fixing it

If you define the problem correctly, you almost have the solution

– Steve Jobs

{CODENATION}

# Research/Brainstorm Ideas...

...**do research** regarding the nature of the problem and possible causes, create **multiple possible solutions**

{CN}®

# Research...

...Might include looking into **causes of similar problems** that have been resolved in the past or **asking those involved in the issue**

{CN}®

Solving problems usually involves creating some **contingency plans** in order to **contain further problems**

{CN}®

**Input** - a goal; research of the problem and possible solutions; imagination

**Output** - pick-list of possible solutions that would achieve the stated goal

{CN}®

# Decide on a Solution...

...The ideal solution will **meet the goal**, efficient, and has the fewest side effects or consequences

# Decision-making skills...

...To find the best solution, **analyse every resolution** and decide which is best for the situation you are in

{CODENATION}

# Elements to consider to narrow down solutions...

...Efficacy, practicality, timeliness, resources, cost, who is involved. Process of elimination also works well. **Discussing with a team** also leads to better results.

**Input** - pick-list of possible solutions; decision making criteria

**Output** - decision of what solution you will implement

# Implement the Solution...

...Implementation requires **planning and execution**. Rushing into it will often lead to a botched solution that doesn't reach the goal

An hour of planning can save you 10 hours of doing

- Dale Carnegie

{CN}®

**Input** - decision; planning; hard work

**Output** - resolution to the problem

{CN}®

# Evaluate the Results...

...Was the goal met? Review what worked, what didn't and what impact the solution had

# Why Evaluate?

...It helps you **improve long-term problem-solving** skills and keeps you from re-inventing the wheel so you will better solve your next problem

**Input** - results of implementation

**Output** - insights on strengths and weaknesses

# Applying It...

...Lets try applying it to an example

# Get Fit...

...You are told by a Doctor to get fit, what do you do?

# Problem: Get Fit

**Define** - I need to Loose weight

**Research/Brainstorm** - Eat healthier, exercise more, start walking/cycling etc.

**Decide on solution** - start eating healthier and cycle to work

**Implementation** - cook healthy meals, cycle

**Evaluate** - Did I loose weight? Can I eat better? Can I walk more in a day as well?

{CODENATION}

# Make a Game...

...What would you need to consider
and what steps would you take?

{CODENATION}

# Problem: Make a Game

**Define** - I want to make a game

**Research/Brainstorm** - Look into genre, platforms, game engines, art style

**Decide on solution** - 2d action/adventure PC game, using Unreal engine

**Implementation** - Plan and Make game

**Evaluate** - Can it be launched? Areas to improve made clear.

{CODENATION}

# Hopefully not so scary now...