

Lecture 01

Lecture contents:

- Course information and contents
- Introduction
- Program design

المحاضرة دي أغلبها كلام مش هيدخل في الامتحان بس هو مهم .. الدكتور قال ان هدف المحاضرة دي اننا نرجع نقراها تاني في اخر الترم عشان نربط الدنيا ببعضها ونعرف أهمية الكورس دة مبدئياً الكورس دة تكمل لكورس ال ++C اللي اخدناه في سنة اولى .. و دة ال course content بتاعنا الترم دة :



Course Contents/Schedule

3

Topic	Lectures
Introduction + Software Engineering + Abstraction	1
C++ Classes (ADT)	3
The List ADT	4-5
Stacks and Queues	2
Searching and Sorting	3
Tables and Fast Searching	2
Trees	2

وفي ال slides هتلاقي شوية لينكات كويسة تساعدنا نعيش مع نفسنا في الكورس

طيب ليه بقي احنا بناخد كورس data structure واحنا مهندسين اتصالات اصلاً؟؟
اول سبب هو انه بيفتح فرص كتير جداً للواحد سواء في مصر او برة مصر لأن ال data structure هو اساس ال programming لأننا قبل اي حاجة بنحتاج نعرف احنا بنتعامل مع ايه .. فهتلاقي ان ال data structure بيفتح لنا فرص في ال embedded systems وفي ال machine learning او ال data science عموماً وفي ال hardware design زي ما عملنا في ال FPGA قبل كدة لما كنا بنتحكم في ال gates اللي فيها ب software، وكمان بتفتح لنا فرص في ال IC design زي ما بنلاقي في شركات زي mentor graphics مثلاً ان هم ليهم software بتاعهم لوحدهم بيشتغلوا بيه .. طب مين اللي عمل ال software دة؟؟

ناس فاهمة كويس data structure وبالتالي ناس قوية في ال programming

الدكتور عايزنا نعتبر اننا بناخد الكورس دة في ورشة .. نبقى عاملين زي بلية الصغير اللي بيقتد يجرب يشيل الحاجة ويشوف ايه التأثير اللي حصل ويرجعها تاني ويجرب لو غير حاجة في ترتيب الحاجات اللي موجودة ايه اللي هيحصل وكدة .. فاحنا كل ما نجرب اكثر كل ما نفهم الدنيا احسن

عشان نعرف أهمية ال software engineers تعالى نبص على اكبر شركات في العالم دلوقتي :

What are the world's most valuable companies?

Ranking of the companies rank 1 to 100	Market value in million U.S. dollars
Apple	752
Alphabet	579.5
Microsoft	507.5
Amazon.com	427
Berkshire Hathaway	409.9
Facebook	407.3
ExxonMobil	242.2
Johnson & Johnson	338.6
JPMorgan Chase	306.6
Tencent Holdings	277.1
Wells Fargo	274.4
Alibaba	264.9
General Electric	261.2

May 2017

بص على Apple هتلاقىها من الشركات القليلة اللي بتشتغل في الهاردوير والسوفتوير مع بعض ودة سبب كبير في انها تكون في المكان اللي هي فيه دلوقتي
بص على Alphabet اللي هي الشركة الأم ل google دلوقتي .. و Microsoft اللي عاملين اشهر OS موجود وكمان دلوقتي بقوا شغالين في ال cloud platform وكل الكلام دة software
وفيسبوك اللي جمع الناس كلها وساعدهم يتواصلوا مع بعض من غير ما يتقابلوا .. و Amazon & Alibaba اللي هم دلوقتي اكبر stores في العالم وكل الناس بتشتري منهم من اي مكان برغم ان مالهمش ولا محل موجود على الارض نقدر نروحله بنفسنا
فال software engineering فتح فرص كثير جداً لأفكار غريبة بتننّفذ وتتجّح نجاح كبير جدا

Software engineering vs Computer science:

- ❑ Computer science focuses on theory and fundamentals
- ❑ Software engineering is concerned with the practicalities of developing and delivering useful software

نبدأ في الكلام المهم بقى ..
لو احنا عايزين نشغل في software project .. اللي احنا بنعمله اننا مش بنشوف الموضوع بحجمه الأساسي فينقول يلا نكتب الكود ونبدأ نكتب اللي بييجي في دماغنا و في النص نكتشف اننا كنا محتاجين نزود حاجة في الاول او نعدل حاجة عملناها فنرجع تاني نزود الحاجة دي ونكمل الكود، نيجي في الاخر نقول ايه دة دة فيه طريقة احسن ممكن اعمل بيها الحاجة دي فنرجع تاني نغير الطريقة .. بس دة اكبر غلط .. احنا كدة بنعمل حاجة اسمها spaghetti code اللي هو الدنيا كلها داخلة في بعضها ومحدث يعرف يقرأ حاجة في الكود او يفهمه .. عشان كدة اول نصيحة هي : **DON'T START BY CODING**

والمفروض نبدأ الاول بال design process .. اللي هي اصلا بتاخذ وقت اكبر من ال development .. ففيه مرحلتين في ال process دي مهمين جدا :

1- Specification:

Answers the question of 'What the system does'.

2- Detailed design:

Describes how the system will achieve the determined specs, so that the system is split into multiple modules

عشان كدة في الشركات او التيمات اللي بتشتغل صح هنلاقي ان فيه حاجة اسمها 4-2-4 time plan .. يعني 4 وحدات من الوقت هيتخصصوا للديزاين (يعني مش في يوم وليلة نقول اننا عملنا ديزاين)، ووحدين بس لل code (لأن الكود يعتبر اسهل حاجة في الموضوع) وبعد كدة 4 وحدات لل testing (يعني برضه مش في يوم وليلة نقول احنا عملنا testing) يعني هنعمله في 10 شهور .. يبقى 4 شهور design وشهرين بنعمل الكود واربع شهور بنعمل testing ... فواضح ان الموضوع مش بالبساطة اللي بنتخيلها وبرضه في نفس الوقت مش بالتعقيد ده ... وساعات الناس بتشتغل ب :

(3-1-6) time plan

بس لو الدنيا مش كويسة والناس ماشية من غير ما تهتم بالديزاين وكدة فالشغل يبقى ماشي بطريقة عشوائية وبالتالي ال testing هياخذ وقت اكبر من كدة بكثير ويمكن اوي المشروع مايكملش في الاخر

بس تعالى هنا .. احنا كدة قسمنا البروجيكت الكبير لكذا واحد صغيرين (كذا module) ودلوقتي المفروض اننا نشغل على كذا بروجيكت بدل ما كنا بنشتغل على project واحد .. ف هل احنا كدة بنعقد الموضوع على نفسنا؟؟
الاجابة لأ مش بنعقد .. احنا لو كنا اشتغلنا على البروجيكت الكبير ده كان هيبقى عندنا نفس ال complexity، بس دلوقتي احنا قسمنا ال complexity دي على كذا مرحلة فبتعامل مع كل مرحلة لوحدها فاحنا في الحقيقة كدة بنسهل الدنيا على نفسنا وخلي بالك كل ما عدد الناس في التيم يزداد كل ما ال complexity بتاعت البروجيكت تزداد .. لأن ال programming مالوش constraints ف كل الناس بتفكر وبتطلع حاجات من دماغها .. فلما التيم بيكبر بنلاقي طرق كتير اوي ممكن نمشي فيها فبنبقى مش عارفين نحدد احنا هنمشي فين بالضبط وبالتالي الدنيا بتتصعب، فالدنيا محتاجة تنقسم كويس الاول عشان مانوصلش للمرحلة دي

طيب ازاى بقى بن manage ال complexity دي؟؟
عن طريق حاجتين:

1- Abstraction

2 – Decomposition (تقسيم المشكلة)

اول حاجة ال abstraction:



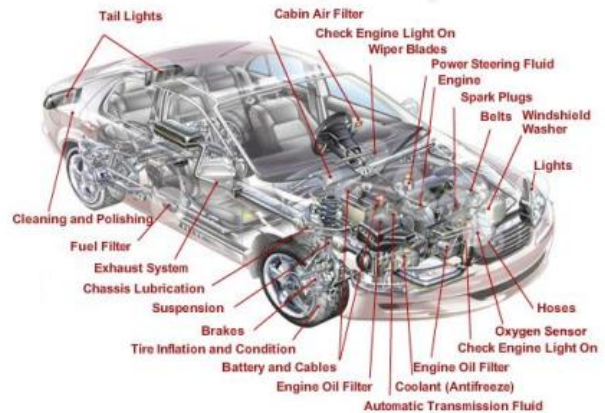
Data Structures == Abstract Data Types

24

- So, what is abstractions all about?
- Abstraction: A model of a complex system that captures only the necessary details.
- Different abstractions are used by different stakeholders of the system.

يعنى ايه برودو abstraction ؟
يعنى الحاجات اللي ببقى محتاج اعرفها عشان اعرف اتعامل بس .. لكن مش لازم تفاصيل .. ناخذ مثال احسن

- Driver cannot know all details of complex working of the car.

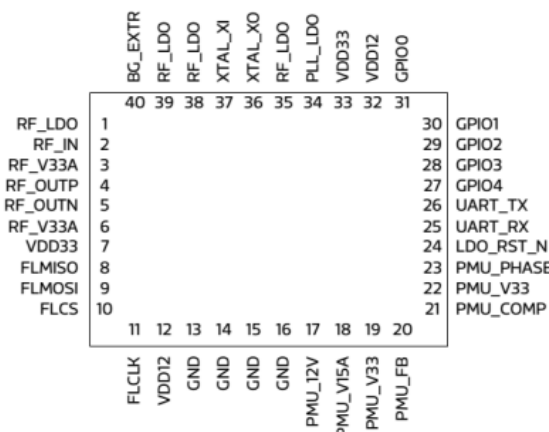


العربية فيها شاسيه و عجلات و متور و stiring system و lighting و locks و brakes
انا كواحد بصمم نظام منهم .. هبقى محتاج طبعا اعرف تفاصيل التفاصيل .. وكل الاحتمالات لما النظام ده يشتغل و و
لكن السواق اللي بيركب العربية بقى .. بيعملها abstraction .. يعنى بيروح يركب و يدور ويمشى .. من غير ما يبقى ايه اللي
بيحصل فى ال engine لما بيدور .. فجأة عربية قدامه وقفت .. فالسواق داس على الفرامل .. مش لازم يفكر ازاى الفرامل بتشتغل.
ولو باظنت هيوديها لمركز الصيانة .. وهناك الفنى هيسأل السواق على المشكلة و يبدأ يشوف السبب.
فكل احد بيتعامل مع العربية بشكل مختلف.

مثال ثانى
لما الدكتور بيدى درجات .. بيدى مثلا على الحضور و الامتحان و اللي بيفضل يجاوب .. مش لازم بقى يعرف سنك و اسم الشهرة
بتاعك و جبت كام فى الثانوية العامة .. ده هو ال abstraction

كمان مثال عشان مبيقالكش حجة D:

- You use the IC by reading datasheet and knowing which each PIN does.
- Then you build a larger system out of this
- This is abstraction
- Another engineering team designs the IC



ال IC جواها hundred of billions of transistors مهندسين كثير اوى اتجمعوا عشان يصمموها.
 انا بقى اما باجى استخدم ال IC دى .. مش لازم اشوف كل اللى جوا ال IC .. انا وصلنى صندوق مقفول طالع منه pins ومعاه زى
 user manual .. مكتوب فيه كل pin يتوصل بيها ايه.. وبرغم انك مش عارف ايه اللى جوا ال IC لكن تقدر بيها تعمل حاجة قيمتها
 اكبر من قيمة ال IC
 ده ال abstraction

كده المحاضرة خلصت .. بس الدكتور قال ان فى exercise فى ال slides وفى homework فى اخر slide
 وقال كمان اننا هنستعمل Eclipse/gnu C++ compiler

بس كده