Nile University

Data Structures and Algorithms

Episode One

Author: Ahmed Khaled

Nile University

Know Your TA

Ahmed Khaled

My Friends << Monier << Asmaa << Rawan << endl

Ain Shams University. Topping my Class

One Semester of Experience at NU

Master Student at NU

I like Teaching, I hate Grading. **we** can help

Material & Code Files

Everything you'll see are There on This Github Repo.

Updated Regularly.

Grading Scheme

- Labs & Attendance/Participation:	10%
- 2 Practical Quizzes	10%
- 2 Assignments (individual)	15%
- 1 Project (group-based):	15%
- Midterm:	20%
- Final:	30%

Agenda

- C++ Quick Review [30 mins]

  Input/Output, Variables, Conditionals,
  Loops, Arrays, Functions,
  Pointers, and Structs.

- OOP Introduction [30 mins]

  Class, Constructor(s), Destructor,
  Attributes (Private & Public), Methods (Getters & Setters),
  Function Overloading

- Hands-on [30 min]

Logistics and Policies

Each Lab (except This!) will begin with a Quiz on Exactly the Lab Start Time for 15 Minutes, Delay on Attending will cost you the Quiz some of or All its Time!

Grades should be Transparent for you, no week are over without declaring your Contribution Marks.

Lab Attendance is Mandatory, but it’s gradeless in itself, Quiz has 100%.

You MUST come with any necessary software Packages & Tools Installed and a Blank Project is ready. Any Time loss because of not being well-prepared will affects your Grade by -1 mark.

Given That … we still wish you All the Best and Enjoy This Journey with Us <3.

C++ Programming Language Review

Let’s Show Not Tell

- Standard Input & Output - Two Thirds of Computing Workflow
- Variables and Data Types - Flexibility and Dynamic Processing
- Conditionals - Branching and make Decision
- Loops and Arrays - Handle Collections of Data
- Functions - Program Modularity, being Composable
- Pointers - Close Memory Management and Dynamic Memory Allocation
- Structs - Compound Data Types [Having Parts]

Nile University

Object-Orientation using C++

Object Oriented Programming

is a way of solving complex problems by breaking them into smaller problems using objects.

OOP Design Principles:-

- Encapsulation
- Polymorphism
- Inheritance
- Abstraction [Information Hiding]

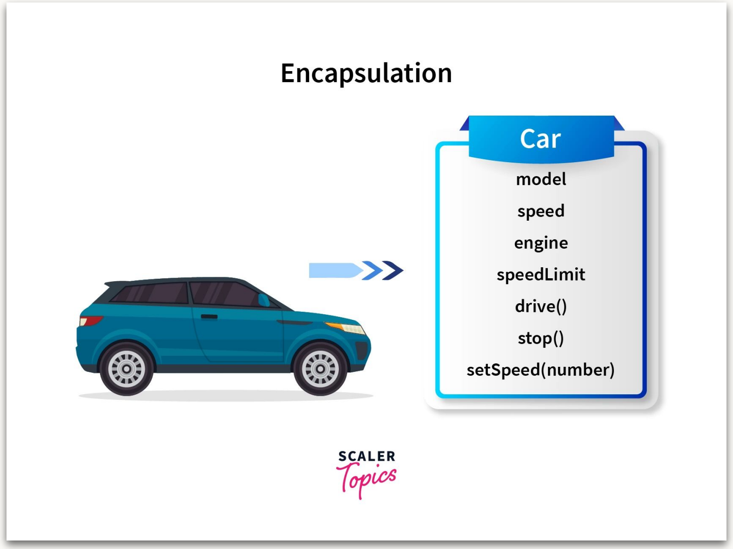
Encapsulation (Abstraction) (Code modularity)

binding together the data and the functions that manipulates them

Then, objects that can interact with each other.

Let’s Code a Car Class with:-

+ speed and model Attributes
+ Two Constructor, One Default and One Parametrized
  Setters and Getters
+ Destructor
+ One Overloaded Member Function,
  How much Time, if it went x miles ?



Polymorphism (Code Reuse)

is a powerful concept in OOP and allows for more flexible and extensible code by promoting code reuse and reducing the amount of duplicate code required.

Will Shine on Inheritance. Next Time.

Key Concepts (Car Example)

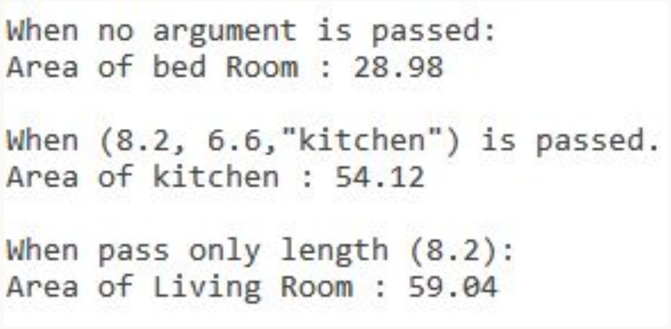
Encapsulation: Keeping the data (speed and model) private and accessing them through public methods.

Polymorphism: Demonstrated by overloading the calculateTime function to handle both integer and double types.

Constructors and Destructor: Used to initialize and clean up objects.

Hands-On

+ Write a program to handle Rooms of any apartment by creating Room Class.
+ Member variables: Length, breadth, room_name.
+ Make three constructors:
  - Default: length = 6.9, breadth = 4.2, room_name = “bed Room”.
  - Parameterized: by passing (Length, breadth, room_name) to member variables.
  - Constructor with passing one argument (Length) and fixed breadth = 7.2 and room_name = “Living Room”.
+ Apply Encapsulation by using Setters and getters.
+ Calculate Area of the Room.
+ Create 3 objects to apply every constructor type.



```
When no argument is passed:
Area of bed Room : 28.98

When (8.2, 6.6,"kitchen") is passed.
Area of kitchen : 54.12

When pass only length (8.2):
Area of Living Room : 59.04
```

Nile University

Thank You