**Week 0 Tasks.**

Day 1

Topics

* Intro to Gaming Industry.
* How Game Engine Works, Libraries vs Engine.
* Memory, Data types, variables, reference type, value type, dry run, operators, if / else statement.

Mandatory Tasks.

* (101) Write a program to check whether the 3 digit integer is palindrome or not?
* (102) Write a program to flip the first and last digit of an integer.
* (103) Write a program which checks whether the given year is palindrome or not.

Day 2

Topics

* Loops/ Types of Loops/ Counter Controlled / Conditional Loops / Nested Loops.

Mandatory Tasks.

* (201) Write a program to calculate the nth power of a number.
* (202) Write a program to check whether a number is prime or not.
* Print the Following Shapes on screen. (203)Triangle, (204) Inverted Triangle, (205)Hollow Square, (206)Hollow Inverted Triangle)
* (207) DryRun of Inverted Triangle discussed in class.

Day 2 Bonus Tasks.

* (208) Print first 20 prime numbers.
* (209) Print a square with a hollow inverted triangle inside.

Day 3

Topics.

* Arrays. How it is stored.
* Functions ( Parameters and Returns )
* Reference Type and Value Type.

Mandatory Tasks.

* (301) Find the Largest number from an array.
* (302) Find the Smallest Number from an array.
* (303) Sort an array with the help of any algorithm
* (304) Write a function to check whether the string is palindrome or not.
* (305) Write a function to print all palindrome subStrings.

Day 4

Topics.

* OOP. Procedural vs Object oriented *(Explaining the need of OOP, access modifiers why we need them)*
* Classes / Objects / Separating Interface and Implementation
* Demo of Car Simulation Project.

Mandatory Tasks.

* (Write an OOP structure with different cars and cars can attack each other, take damage, move, and get Destroyed)
* The Attack Function could be overridden.

Day 5/6

Topics

* Inheritance / Polymorphism / Types of Polymorphism./ Composition / Design Patterns / Factory Patterns.

Mandatory Tasks.

* Procedural vs Object oriented *vs reactive (Explaining the need of OOP, access modifiers why we need them)*
  + Create a player who has 5 types of gun but write code that should be extendable. We can add more guns in the future without changing the previous code.
  + Create a game in which we have x type of players and they have n number of abilities. Like stun, heal, smoke, flash etc
  + Create an alarm system when a specific time reaches all user guns should be changed to basic ammos.
  + Leaderboard of the player who has the most kills and it should update simultaneously no graphical user interface is required.
  + Inventory system: The system should incorporate future guns and those guns should be equippable by the player.

Addition:

Write a program to mirror a binary tree.

# Unity Coroutines: How Do They Work?