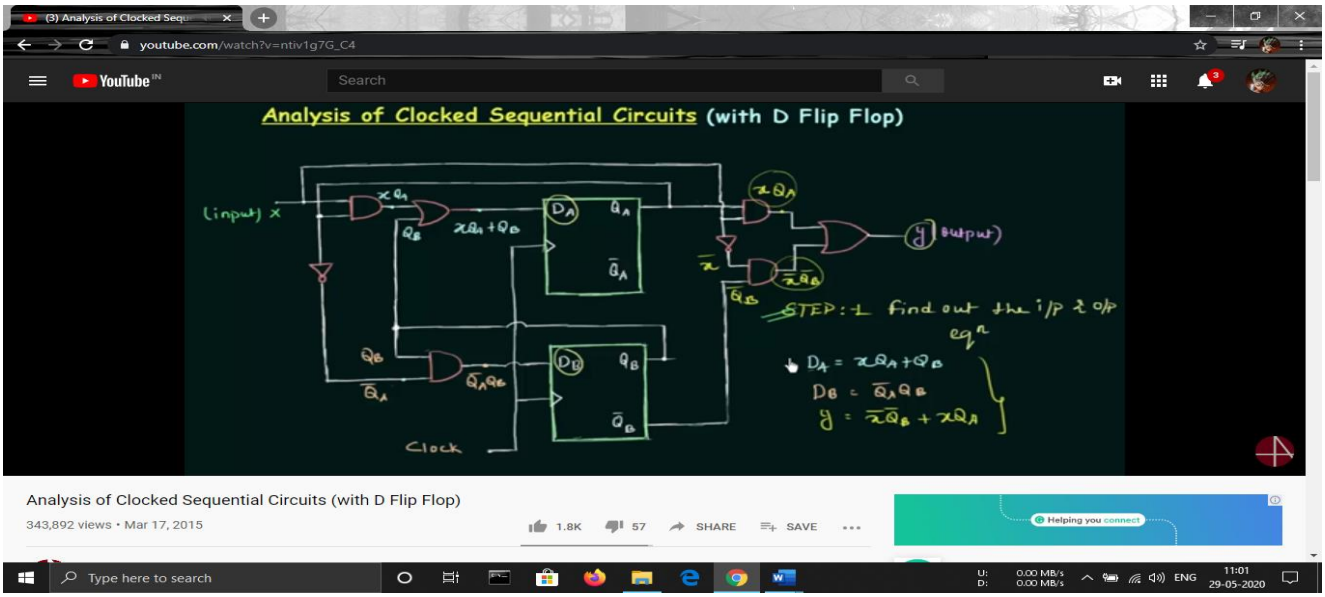
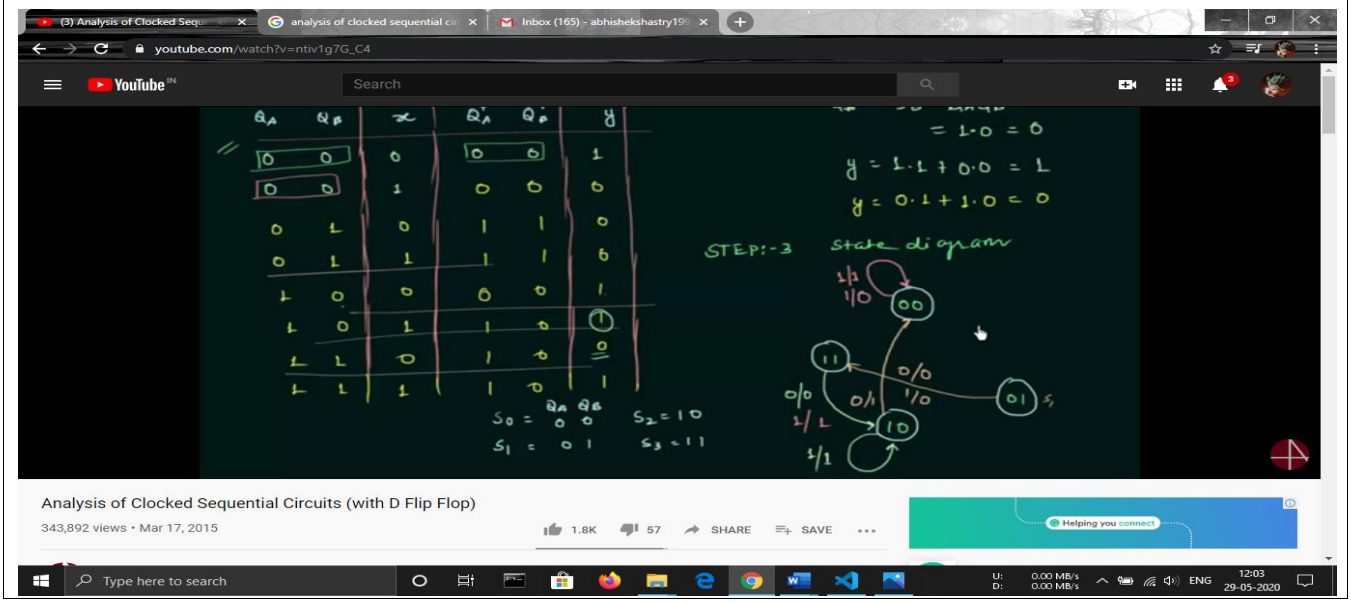
DAILY ASSESSMENT REPORT

Date:	**29/05/2020**	**Name:**	**Abhishek M Shastry K**
Subject:	**Logic Design**	**USN:**	**4AL17EC002**
Topic:	**1] Analysis of clocked sequential circuits 2] Digital clock design**	**Semester & Section:**	**6th 'A'**
Github Repository:	**AbhishekShastry-Courses**		

FORENOON SESSION DETAILS

Image of session





Report

Analysis of clocked sequential circuits

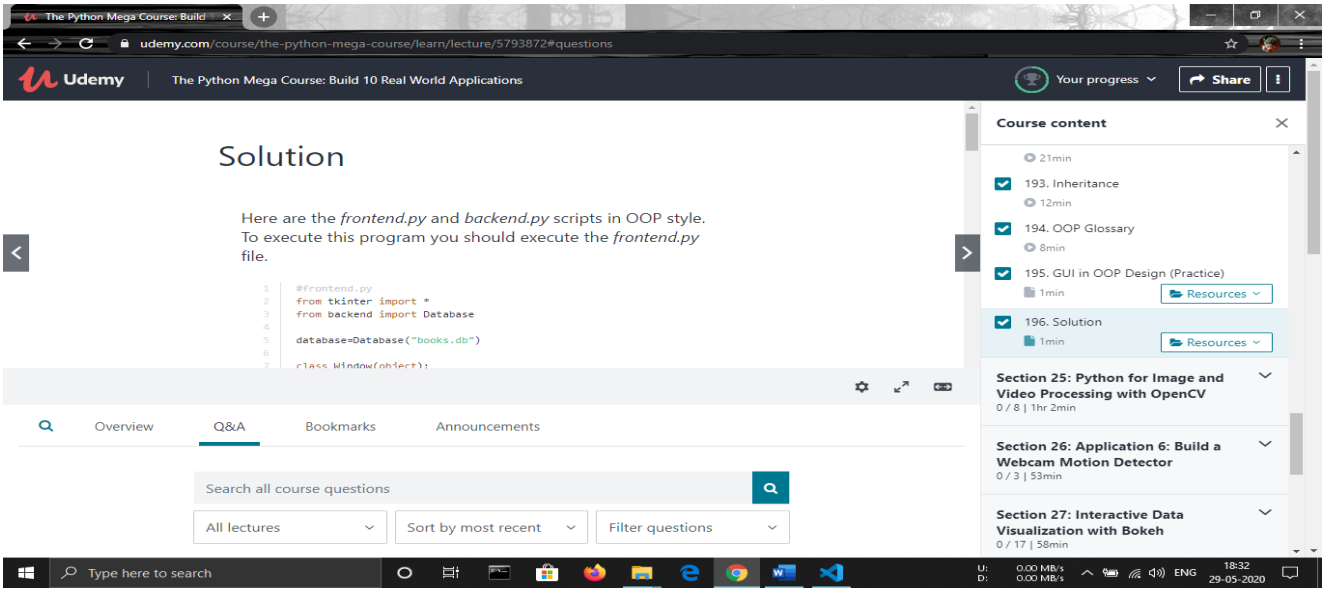
A sequential circuit is the assimilation of a combinational logic circuit and a storage element. With the applied inputs to the combinational logic, the circuit outputs are derived. These sequential circuits deliver the output based on both the current and previously stored input variables. The derived output is passed on to the next clock cycle. Sequential circuits consist of memory devices to store binary data. This binary information describes the current state of the sequential circuit.

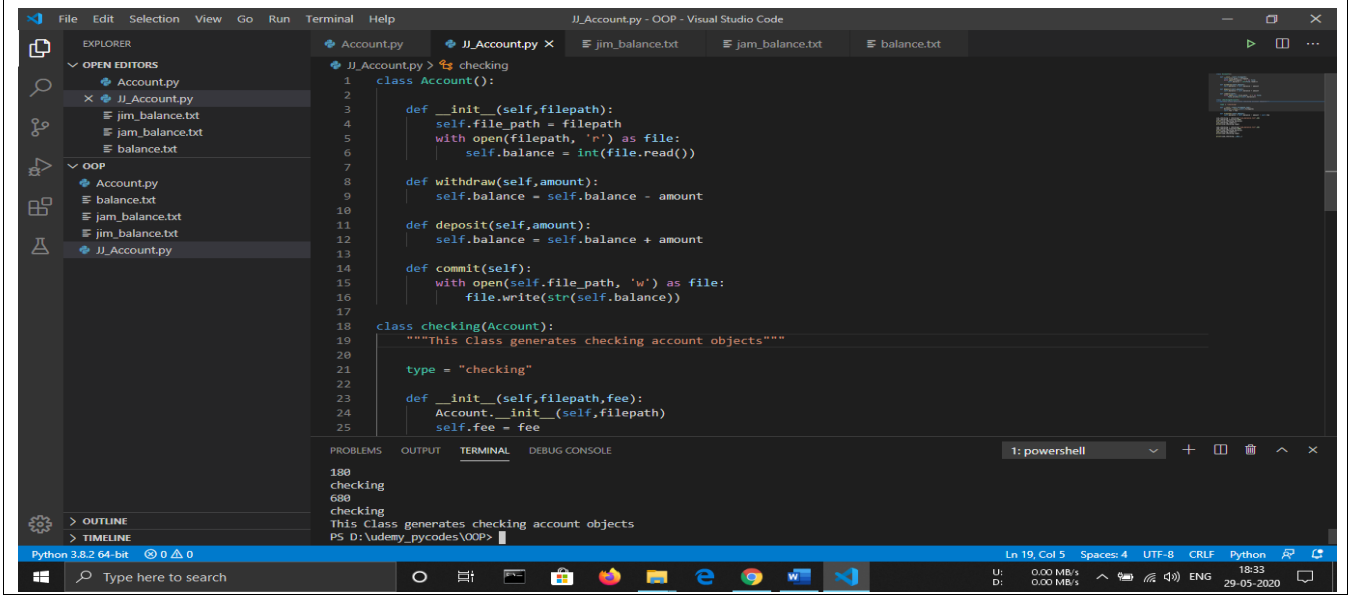
- In the clocked sequential circuit two D flipflops are used along with the logic gates for input combinational (x) and output combinational (y).
- There are 3 steps to obtain state diagram for clocked sequential circuit.
- Step 1: Find the input and output equation.
    - ✓ Find the input expression for D_A and D_B.
    - ✓ Find the output expression for y.
- Step 2: State Table - The state table representation of a sequential circuit consists of three sections labeled present state, next state and output. The present state designates the state of flip-flops before the occurrence of a clock pulse. The next state shows the states of flip-flops after the clock pulse, and the output section lists the value of the output variables during the present state.
    - ✓ First column consists of present state Q_A and Q_B.
    - ✓ Second column consists of input x.
    - ✓ Third column consists of next state Q_A^+ and Q_B^+.
    - ✓ The last column consists of the output y.
- Step 3: State diagram - In addition to graphical symbols, tables or equations, flip-flops can also be represented graphically by a state diagram. In this diagram, a state is represented by a circle, and the transition between states is indicated by directed lines (or arcs) connecting the circles.
    - ✓ Since there are 2 flipflops used, the 4 possible states are given by S_0, S_1, S_2 and S_3.
- Applications of sequential circuits.
    - ✓ Used as registers inside microprocessors and controllers to store temporary information.
    - ✓ Applied in programmable devices such as CPLD, PLD, and FPGA.

Date:	29/05/2020	Name:	Abhishek M Shastry K
Course:	The Python Mega Course: Build 10 Real World Applications	USN:	4AL17EC002
Topic:	1] Object Oriented Programming	Semester & Section:	6th 'A'
Github Repository:	AbhishekShastry-Courses		

AFTERNOON SESSION DETAILS

Image of session





Report

Object Oriented Programming

- One of the popular approaches to solve a programming problem is by creating objects. This is known as Object-Oriented Programming (OOP).
- The concept of OOP in Python focuses on creating reusable code. This concept is also known as DRY (Don't Repeat Yourself).
- In Python, the concept of OOP follows some basic principles:
  - ✓ Inheritance - A process of using details from a new class without modifying existing class.
  - ✓ Encapsulation - Hiding the private details of a class from other objects.
  - ✓ Polymorphism - A concept of using common operation in different ways for different data input.
- Some terminologies in OOP:
  - ✓ Class - A class is a blueprint for the object which contains all the details about the object.
  - ✓ Object - An object (instance) is an instantiation of a class. When class is defined, only the description for the object is defined. Therefore, no memory or storage is allocated.
  - ✓ Methods - Methods are functions defined inside the body of a class. They are used to define the behaviors of an object.
  - ✓ Inheritance - Inheritance is a way of creating new class using the details of existing class without modifying it and extra functions can also be added to the derived class.
  - ✓ Data member - A class variable or instance variable that holds data associated with a class and its objects.
  - ✓ Function overloading - The assignment of more than one behavior to a particular function. The operation performed varies by the types of objects or arguments involved.
  - ✓ Instantiation - The creation of an instance of a class.
  - ✓ Operator overloading - The assignment of more than one function to a particular operator.

- Class attributes are variables of a class that are shared between all of its instances. They differ from instance attributes in that instance attributes are owned by one specific instance of the class only, and are not shared between instances.
- Some of the built-in class attributes:
    - ✓ **"__init__"** is a reserved method in python classes. It is called as a constructor in object-oriented terminology. This method is called when an object is created from a class and it allows the class to initialize the attributes of the class.
    - ✓ The **__del__()** method is a known as a destructor method in Python. It is called when all references to the object have been deleted i.e. when an object is garbage collected.
    - ✓ Python objects have an attribute called **__doc__** that provides a documentation of the object.
- Class variables are defined within the class construction. Because they are owned by the class itself, class variables are shared by all instances of the class. They therefore will generally have the same value for every instance unless you are using the class variable to initialize a variable.
- Instance variables are owned by instances of the class. This means that for each object or instance of a class, the instance variables are different. Unlike class variables, instance variables are defined within methods.
- The main advantage of OOP is that it reduces the number of lines in the code and also makes the code more readable.
- Client-server systems, Object-oriented database, Real-time system design, etc. are some of the applications of OOP.