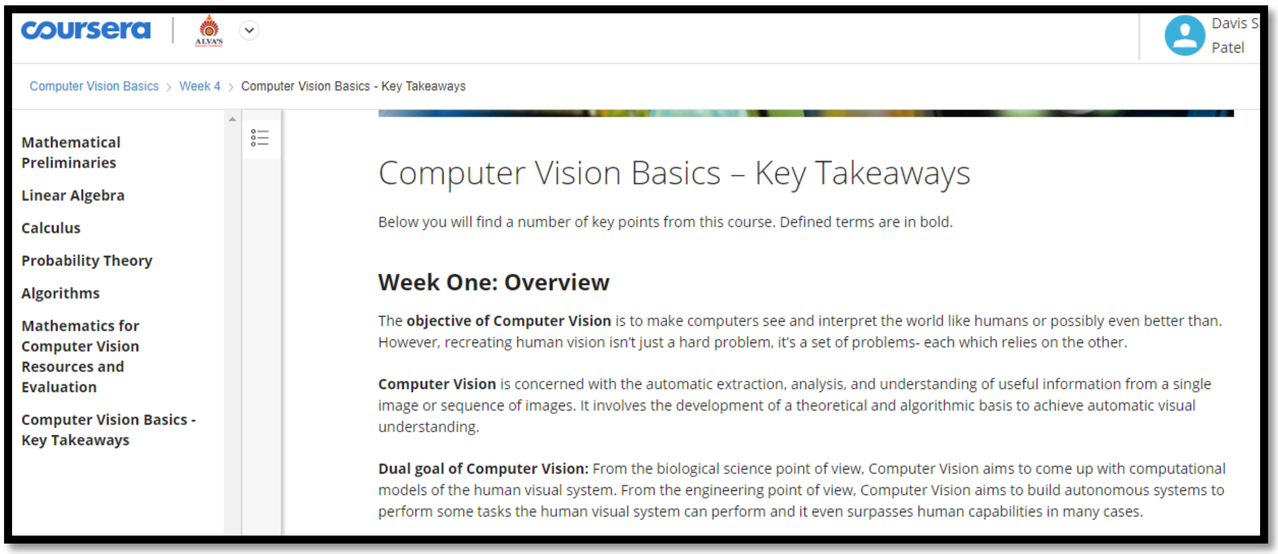
DAILY ASSESSMENT

Date:	17/07/2020	Name:	Davis S. Patel
Course:	Computer Vision Basics	USN:	4AL16EC045
Topic:	Computer Vision Basics - Key Takeaways	Semester & Section:	8th - A
GitHub Repository:	Davis		

FORENOON SESSION DETAILS

Image of session


coursera
Davis S Patel
Computer Vision Basics > Week 4 > Computer Vision Basics - Key Takeaways
Mathematical Preliminaries
Linear Algebra
Calculus
Probability Theory
Algorithms
Mathematics for Computer Vision
Resources and Evaluation
Computer Vision Basics - Key Takeaways

Computer Vision Basics – Key Takeaways

Below you will find a number of key points from this course. Defined terms are in bold.

Week One: Overview

The **objective of Computer Vision** is to make computers see and interpret the world like humans or possibly even better than. However, recreating human vision isn't just a hard problem, it's a set of problems- each which relies on the other.

Computer Vision is concerned with the automatic extraction, analysis, and understanding of useful information from a single image or sequence of images. It involves the development of a theoretical and algorithmic basis to achieve automatic visual understanding.

Dual goal of Computer Vision: From the biological science point of view, Computer Vision aims to come up with computational models of the human visual system. From the engineering point of view, Computer Vision aims to build autonomous systems to perform some tasks the human visual system can perform and it even surpasses human capabilities in many cases.

REPORT –

The **objective of Computer Vision** is to make computers see and interpret the world like humans or possibly even better than. However, recreating human vision isn't just a hard problem, it's a set of problems- each which relies on the other.

Computer Vision is concerned with the automatic extraction, analysis, and understanding of useful information from a single image or sequence of images. It involves the development of a theoretical and algorithmic basis to achieve automatic visual understanding.

Dual goal of Computer Vision: From the biological science point of view, Computer Vision aims to come up with computational models of the human visual system. From the engineering point of view, Computer Vision aims to build autonomous systems to perform some tasks the human visual system can perform and it even surpasses human capabilities in many cases.

Computer Vision systems contain these basic elements:

1. A power source
2. At least one camera
3. A processor
4. Control and communication cables or some kind of wireless interconnection mechanism
5. Configurable software
6. A display in order to monitor the system

Some of the applications of Computer Vision include: facial recognition in smartphone cameras, analysis of surroundings in self-driving cars, and factory robots that navigate around coworkers.

The field of Computer Vision heavily incorporates concepts from the areas of digital processing, neuroscience, computer graphics, solid-state physics, photogrammetry, and artificial intelligence.

The field of **Digital Image Processing** predominantly deals with image-to-image transformations. Typical image processing operations include image compression, image restoration, and image enhancement.

Computer Graphics studies the techniques that produce image data from 3D models, whereas computer vision works to produce 3D models from image data.

Machine Vision is the process of applying a range of technologies and methods to provide imaging-based automatic inspection, process control, and robot guidance in industrial applications.

Photogrammetry is the science of making measurements from photographs, especially for recovering the exact positions of surface points captured in the image.

The field of Computer Vision emerged in the 1950s, with research along three distinct lines: replicating the eye, replicating the visual cortex, and replicating the rest of the brain.

Optical Character Recognition (OCR) makes typed, handwritten, or printed text intelligible for computers.

Convolutional Neural Networks (CNNs) have been applied to identify faces, objects, and traffic signs, as well as powering vision in robots and self-driving cars.

Computer Vision is outperforming humans on certain restricted real world tasks such as circuit board inspections and facial recognition under controlled conditions.

A **point light source** originates at a single location in a 3 dimensional space, like a small light bulb, or potentially at infinity like the sun.

The key factors that affect the “color” of a pixel are:

1. The light sources

2. Object surface properties

3. Emittance and reflective spectrum

4. Relative position and orientation

The most basic camera model is a **pinhole camera model**. In this model, conceptually, all light passes through a vanishingly small pinhole placed at the origin and illuminates an image plan beneath it. When using a pinhole camera model, this geometric mapping from 3D to 2D is called a **perspective projection**.

Perspective projection makes parallel lines in the real world appear that they might be converging. The point of convergence is called a **vanishing point**.

Three levels in David Marr's paradigm:

1. *Computational theory* – describes what the device is supposed to do

2. *Representation and algorithm* – addresses precisely how the computation may be carried out

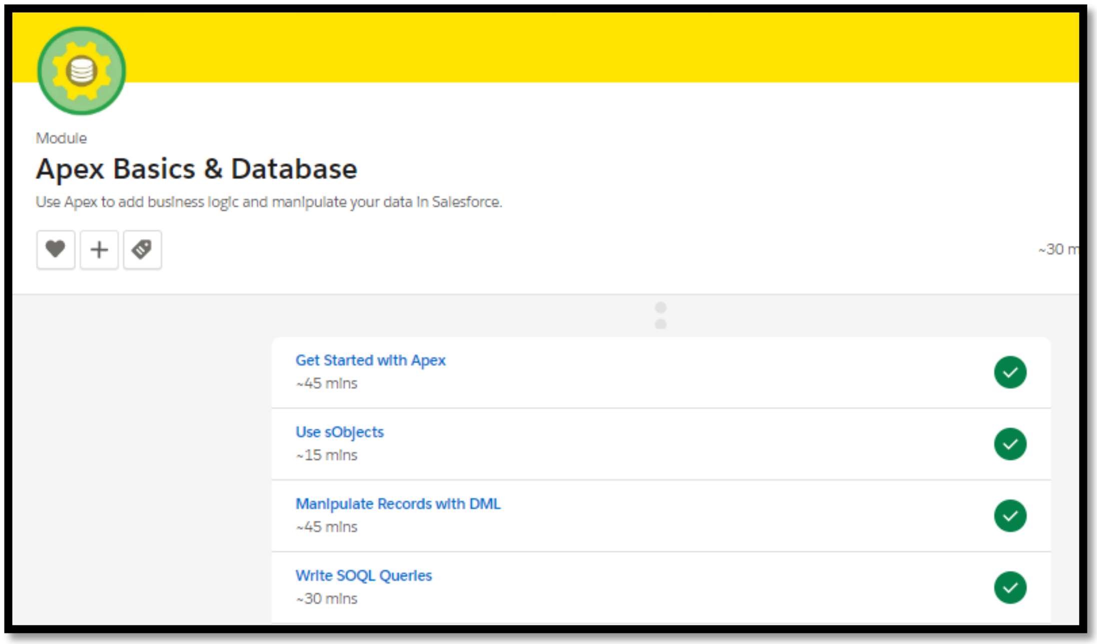
3. *Implementation* – includes physical realization of the algorithm, programs, and hardware

DAILY ASSESSMENT

Date:	17/07/2020	Name:	Davis S. Patel
Course:	Salesforce Developer	USN:	4AL16EC045
Topic:	Apex Basics & Database	Semester & Section:	8th - A
GitHub Repository:	Davis		

AFTERNOON SESSION DETAILS

Image of Session



Module

Apex Basics & Database

Use Apex to add business logic and manipulate your data in Salesforce.

~30 m

Get Started with Apex
~45 mins

Use sObjects
~15 mins

Manipulate Records with DML
~45 mins

Write SOQL Queries
~30 mins

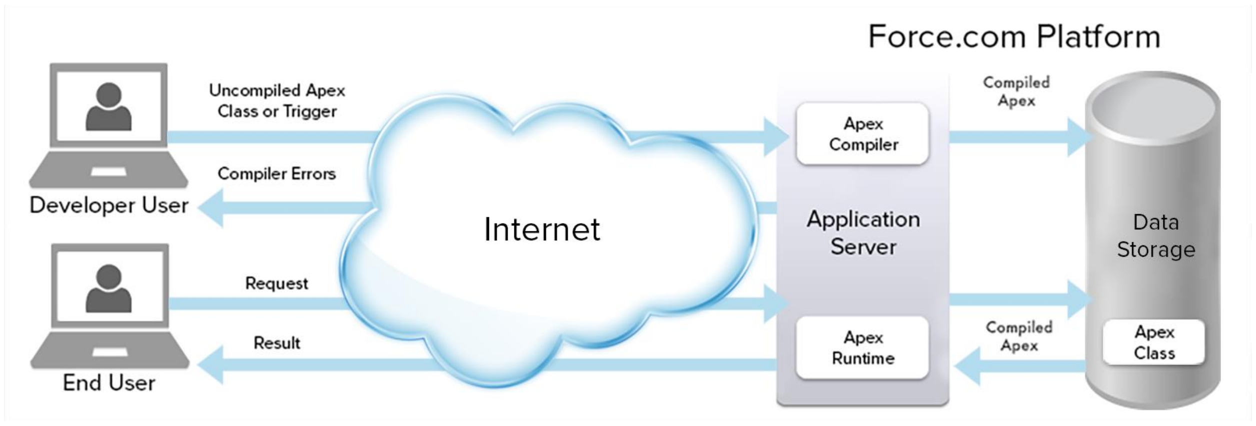
REPORT –

Get Started with Apex

Apex is a programming language that uses Java-like syntax and acts like database stored procedures. Apex enables developers to add business logic to system events, such as button clicks, updates of related records, and Visualforce pages.

As a language, Apex is:

- Hosted—Apex is saved, compiled, and executed on the server—the Lightning Platform.
- Object oriented—Apex supports classes, interfaces, and inheritance.
- Strongly typed—Apex validates references to objects at compile time.
- Multitenant aware—Because Apex runs in a multitenant platform, it guards closely against runaway code by enforcing limits, which prevent code from monopolizing shared resources.
- Integrated with the database—It is straightforward to access and manipulate records. Apex provides direct access to records and their fields, and provides statements and query languages to manipulate those records.
- Data focused—Apex provides transactional access to the database, allowing you to roll back operations.
- Easy to use—Apex is based on familiar Java idioms.
- Easy to test—Apex provides built-in support for unit test creation, execution, and code coverage. Salesforce ensures that all custom Apex code works as expected by executing all unit tests prior to any platform upgrades.
- Versioned—Custom Apex code can be saved against different versions of the API.



Apex Language Highlights

Like other object-oriented programming languages, these are some of the language constructs that Apex supports:

- Classes, interfaces, properties, and collections (including arrays).
- Object and array notation.
- Expressions, variables, and constants.
- Conditional statements (if-then-else) and control flow statements (for loops and while loops).

Unlike other object-oriented programming languages, Apex supports:

- Cloud development as Apex is stored, compiled, and executed in the cloud.
- Triggers, which are similar to triggers in database systems.
- Database statements that allow you to make direct database calls and query languages to query and search data.
- Transactions and rollbacks.
- The global access modifier, which is more permissive than the public modifier and allows access across namespaces and applications.
- Versioning of custom code.

In addition, Apex is a case-insensitive language.

Apex supports various data types, including a data type specific to Salesforce—the sObject data type.

Apex supports the following data types.

- A primitive, such as an Integer, Double, Long, Date, Datetime, String, ID, Boolean, among others.
- An sObject, either as a generic sObject or as a specific sObject, such as an Account, Contact, or MyCustomObject__c (you'll learn more about sObjects in a later unit.)
- A collection, including:
    - A list (or array) of primitives, sObjects, user defined objects, objects created from Apex classes, or collections
    - A set of primitives
    - A map from a primitive to a primitive, sObject, or collection
- A typed list of values, also known as an ***enum***
- User-defined Apex classes
- System-supplied Apex classes