

# Geography 485L/585L Goals, Objectives and Readings

Internet mapping technologies are an important component of geospatial data capture, sharing, visualization, and delivery. This course provides a survey of current and emerging internet and geospatial interoperability standards, technologies, and capabilities. The emphasis of the work in this class will be hands-on experience in four critical aspects of Internet-enabled mapping:

- The basic concepts behind web mapping technologies that enable the delivery of maps and mapped data through web browsers
- The Open Standards that facilitate the exchange of map images and geospatial data over the internet
- The use of published standards-based services in desktop mapping applications that implement those standards
- The deployment of standards-based geospatial map and data services that other systems and users may make use of

The specific class objectives that relate to these activities and departmental curriculum objectives for undergraduate and graduate students in the Geography Department include the following:

- Students will understand the concepts geospatial data and service interoperability
- Students will be able to define the specific requirements of a particular analysis or project and identify the interoperability standards that are capable of meeting those requirements
- Students will be knowledgeable in the core technologies that they may use to produce their own internet-enabled mapping capabilities
- Students will understand the strengths and limitations of current internet mapping technologies for generating cartographically effective map products.

The weekly goals, objectives and reading assignments are outlined below

## Week 1 - Introductions, Course Outline and Introduction to Web and Internet Concepts

This week we will review the content and structure for the course and spend some time getting to know each other.

### Weekly Milestone - Creating Your First Web Page in GitHub

This week's milestone is to set up your GitHub account so you can get started in creating your web portfolio that will be incrementally built over the course of the next sixteen weeks. As part of the process of getting all the parts and pieces into place you will create your first web page within the GitHub environment.

[Link to this week's milestone](#)

### Readings

- [Computer History Museum - Internet History Exhibit Timeline](#)
- [Wikipedia article - History of the World Wide web](#)
- *HTML Manual of Style* - Chapter 1

## Week 2 - Module 2.1 - Web-based Mapping Clients: Introduction to HTML, CSS, and Javascript

### Background

This week we will begin to build our foundation for developing material to be shared over the Internet via the World Wide Web. In particular we will cover the basic process of web development, define the parts of a web page, and spend some time learning about the different *languages* and define the key components of a web page: its structure, presentation, and behavior.

The presentation of information over the Internet is dependent upon the use of standards that have been developed for defining the *structure*, *presentation*, and *behavior* of content. This week we will begin working with the key technologies that define these three components of web content.

These concepts will be illustrated through reference to several simple web pages which are progressively modified to integrate all three of these components.

### Expected Outcomes

By the end of this class module you should understand the following:

- The basic process of web development
- The parts of a web page
- The role of the three web page components: *structure*, *presentation*, and *behavior*
- Be able to write your own basic web page with your own content and make it available over the web

### Key Concepts

- Parts of a web page
- Structure = X/HTML
- Presentation = CSS
- Behavior = Javascript
- Iterative Development

### Weekly Milestone - Create a More Complex Web Page and Style It

[Link to this week's milestone](#)

### Readings

- *HTML Manual of Style* - Chapter 2
- *Designing with Web Standards - Before You Begin*, Chapter 1, Chapter 2 (*optional*)

## Week 3 - Module 2.2 - Web-based Mapping Clients: Google Maps API - Part I

### Background

This week we will begin our work with the popular Google Maps *Application Programming Interface* (API) in developing an interactive web-based mapping client. This development activity will build upon the the work you've done over the last couple of weeks in developing basic web pages by using the capabilities that Google has made available for building mapping interfaces based upon their Maps platform. You will begin working with javascript as a client programming language to both interact with Google's servers and to provide the needed information for Google's mapping tool in your web page.

### Expected Outcomes

By the end of this class module you should understand the following:

- What an Application Programming Interface (API) is
- How Javascript can be used to define the behavior of elements in a web page
- What the basic structure of a javascript code block for defining a Google Maps - enabled page looks like
- How to write a basic web page that includes an interactive Google Map

### Key Concepts

- Application Programming Interface (API)
- Javascript and its location within an HTML page
- The interaction between javascript behaviors and structural elements in a web page

### Weekly Milestone - Creation of a Web Page with an Embedded Google Map

[Link to this week's milestone](#)

### Readings

- *HTML Manual of Style* - Chapter 3
- *Google Maps API* Chapters 1-3 (skim chapter 2)
- [Google Maps API Tutorial](#)

## Week 4 - Module 2.2 - Web-based Mapping Clients: Google Maps API - Part II - Related Topics

### Background

This week covers some additional topics related to the Google Maps API, particularly focusing on styling the Maps base maps using the [styled maps wizard](#) and integrating the javascript generated by the wizard into the base web page code developed last week; and, using [Google's Fusion Tables](#) tool to create and manage tabular data for mapping and other visualization. We complete our work with the Maps API with an example of a more "real" example of a maps-enabled web page.

## Expected Outcomes

By the end of this class module you should be able to:

- Generate a Google Maps JSON style using the *Styled Maps Wizard*
- Integrate that JSON into your map client page for styled basemap display

You should also understand

- The potential of *Fusion Tables* as an alternative source of data to integrate into a custom Google Map page
- The potential structure of an *operational* web page, including the physical separation of page components (structure, presentation, behavior) into separate files

## Key Concepts

- Generating Google Maps styles
- Integrating styles into a Google Maps page
- Fusion tables as a data source for Google Maps maps
- Separation of structure, presentation, and behavior in web development

## Weekly Milestone - Styling of an Embedded Google Map

### Deep Dive - Creation of a a Google Maps Web Page with Custom Points and Labels

## Readings

- [Google Maps API Tutorial](#) (reference)
- [Google Maps Fusion Mapper](#)
- *Google Maps API* Chapters 4-8 (skim for reference)

## Week 5 - Module 3 - GIS and Services Oriented Architectures (SOA)

### Background

Core the the development of distributed mapping systems over the internet is the concept of web services and the interoperability upon which they are based as the means of communication between systems. This week's lecture and focuses on the core concepts of geospatial *Services Oriented Architectures* and the open interoperability standards from the *Open Geospatial Consortium* that enable the exchange of map images and data over the web.

## Expected Outcomes

By the end of this class module you should understand the following:

- The difference between raster and vector data formats and the utilities for retrieving information about supported file formats
- The available resources for determining the projection parameters for different named coordinate reference systems and the command line tools for performing basic coordinate transformations
- The three general tiers of a geospatial services oriented architecture and the components that may exist in those tiers
- The key Open Geospatial Consortium standards for access, data, and representation

## Key Concepts

- Raster and Vector Data Models
- Basic concepts of coordinate transformation and how the parameters that define a transformation may be used to perform basic coordinate transformation calculations using the `cs2cs` utility
- The tiers of a geospatial services oriented architecture
- The constituent components of SOA tiers
- The role of OGC services in providing connectivity between SOA tiers
- The OGC WMS, WFS, WCS, GML, and KML standards and their respective capabilities and purposes

## Weekly Milestone - Proj4, GDAL, and OGR

### Readings

*For Reference*

- [GDAL Utilities documentation](#): `gdalinfo`, `gdal_translate`, `gdaladdo`, `gdalwarp`
- [OGR Utilities documentation](#): `ogrinfo`, `ogr2ogr`
- Proj4 Documentation: The various links under the “Documentation” section of the Proj4 [Project Page](#)

## Week 6 - Module 4.1 - Interoperability Standards - XML, KML, and WMS

### Background

This week’s class focuses on three open interoperability standards that are the most broadly used of the standards that we will be covering.

**Extensible Markup Language (XML)** The World Wide Web Consortium (W3C) standard that is the foundation for many other service and data standards including: the service metadata (GetCapabilities) for the OGC WMS, WFS, and WCS, Geography Markup Language (GML), and KML.

**KML** Formerly known as Keyhole Markup Language, an OGC standard since 2008, KML is a combined geospatial data and representation standard that enables the combined transfer of both location-based data and styling information within a defined XML model.

**Web Map Service (WMS)** The OGC standard for providing on-demand map visualizations based upon user provided parameters reflecting selected data layers, defined areas of interest, image formats, and optionally time of interest.

## Expected Outcomes

At the end of this class students should have an understanding of the following:

- The basic characteristics of XML documents, including the concepts of *well-formed* and *valid* XML
- The capabilities of KML for providing both data and representation information for geospatially referenced data.
- The request-response model for OGC WMS, including the required and optional request parameters for the *GetCapabilities*, *GetMap*, and *GetFeatureInfo* requests; and the response types generated in response to those requests.
- A general familiarity with the linkage between the WMS and KML standards

## Key Concepts

- XML as a general standard for structured data exchange, with DTDs and Schemas defining application specific data models
- KML as a data and representation standard for delivery of geospatial data and symbolization information into client applications, both desktop and web-based.
- WMS as a geospatial data visualization standard for providing online access to map images in a variety of formats for integration into desktop and web-based mapping applications

## Weekly Milestone - WMS & KML

### Deep Dive - OGC Service Concepts

### Readings

- [OGC Workshop White Paper](#)
- [OGC WMS Implementation Specification](#)
- [OGC KML](#)
- [Google Code KML Documentation](#)

## Week 7 - Module 4.2 - Interoperability Standards - WFS & WCS

### Background

This week's class focuses on two other key Open Geospatial Consortium standards that were created to enable access to geospatial *data* of a variety of types.

**Web Feature Service (WFS)** A standard designed for providing on-demand access to features (typically points, lines, and polygons and more complex combinations of these feature types) and their associated attributes, in a variety of formats, and optionally filtered by spatial and other query parameters.

**Web Coverage Services (WCS)** A standard focused on providing access to *coverages* representing a variety of data types, but particularly optimized for dynamic delivery of data based upon multi-dimensional gridded data.

### Expected Outcomes

At the end of this class students should have an understanding of the following:

- The request-response model for OGC WFS, including the required and optional request parameters for the *GetCapabilities*, *DescribeFeatureType*, and *GetFeature* requests; and the response types generated in response to those requests.
- The request-response model for OGC WCS, including the required and optional request parameters for the *GetCapabilities*, *DescribeCoverage*, and *GetCoverage* requests; and the response types generated in response to those requests.
- An understanding of the distinction between WMS, WFS, and WCS for use in different usage scenarios (e.g. map images, vector, raster data access)

### Key Concepts

- OGC WFS as a data access standard for *features* and their attributes with support for a variety of query methods including spatial and parameter values.
- OGC WCS as a data access standard for *coverages* representing spatio-temporal gridded data represented in 1- or more dimensions.

### Weekly Milestone - WMS and KML in Google Earth

### Readings

*For Reference*

- [OGC WFS Implementation Specification][<http://www.opengeospatial.org/standards/wfs>]
- [OGC WCS Implementation Specification][<http://www.opengeospatial.org/standards/wfs>]

## **Week 8 - Mid-term Exam and Portfolio review**

## **Week 9 - Spring Break**

## **Week 10 - Module 2.3 - Web-based Mapping Clients: OpenLayers Javascript Framework - Part I**

### **Background**

The Google Maps API provides one method for presenting an interactive mapping tool within a web browser, but there are restrictions for free use based upon Google's [license](#) agreement, and the API is completely controlled by Google - changes are limited to those that Google enables. The [OpenLayers](#) Javascript framework which (as quoted from the project home page)

has been developed to further the use of geographic information of all kinds. OpenLayers is completely free, Open Source JavaScript, released under the 2-clause BSD License (also known as the FreeBSD).

Given its Open Source model, OpenLayers is managed as a community software project, with the development of specific capabilities driven by particular project or functionality needs that come out of the community.

This week's class focuses on the basics of designing an OpenLayers interactive mapping client, including

- The required structural (HTML) and behavioral (Javascript) components
- Examples of creating and adding layer objects to a map
- Examples of a variety of base maps that can be added to a map
- The controls that may be added to a map, and the methods for managing and positioning those controls

### **Expected Outcomes**

At the end of this class the students will be able to:

- Create a new web page that includes an interactive OpenLayers mapper
- Define one or more base map layers as part of the mapper
- Define an appropriate map center and zoom level for the desired map
- Enable and position controls within the map

### **Key Concepts**

At the end of this class students will understand that

- OpenLayers is a Javascript framework that enables web-based interactive mapping
- The OpenLayers framework supports the integration of a variety of proprietary and open source base map services
- Map size, center, zoom level, and layers may all be defined through the Javascript API
- A wide variety of map controls (informational and interactive) may be added to maps

### **Weekly Milestone - OpenLayers Mapping**

#### **Readings**

- OpenLayers Beginner's Guide: Chapters 1, 3, 5 (review for reference)



## **Week 11 - Module 2.3 - Web-based Mapping Clients: OpenLayers Javascript Framework - Part II**

### **Background**

As we learned in our introduction to OpenLayers in last week's class, it provides a flexible platform for integrating geospatial data from a variety of sources in a single interactive mapping environment. These capabilities include data integration from OGC Web Map Services and a wide variety of data formats for display in the mapper. This week's class provides an overview of the options for the OpenLayers **Map** and **Layer** objects, and specific information about configuring OGC WMS, KML, and Vector Geometry Layers and a brief overview of styling of vector data layers.

### **Expected Outcomes**

At the end of this class, students should be able to:

- Create a new map object
- Add a WMS layer to that map
- Add a KML layer to that map
- Add a variety of Vector Geometry objects to the map
- Define styles for the created KML and Vector Feature objects.

### **Key Concepts**

At the end of this class students will understand that

- There are a wide variety of options that may be provided when creating **Map** and **Layer** objects within OpenLayers
- WMS layers may be added to a map from diverse online servers
- KML files may be added to map, but their potential size can pose a challenge to effective integration into online mapping applications
- A wide variety of geometric objects may be defined and added to a map
- There exist a variety of styling options for vector data that are added to a map

### **Weekly Milestone - A Customized OpenLayers Mapping Client**

#### **Deep Dive - Problem Definition and Data Acquisition**

#### **Readings**

- OpenLayers Beginner's Guide: Chapters 4, 6, 7, 10 (review for reference)

## **Week 12 - Module 4.3 - Interoperability Standards - Desktop GIS Integration**

### **Background**

As we've discussed the components of the client tier of our tiered geospatial services oriented architecture we have concentrated on the open standards that can support client applications and the web-based clients that can consume them. Desktop GIS applications can also consume standards-based services, specifically OGC

services. This week's class concentrates on the methods for integrating OGC services into two GIS client applications Quantum GIS and ArcGIS, demonstrating the utility of using external standards-based services as a data and map image source within desktop applications.

### **Expected Outcomes**

At the end of this class, students should be able to:

- Add a WMS service to Quantum GIS
- Add a WFS service to Quantum GIS
- Add WMS, WFS, and WCS services to ArcGIS (if they have access to the required software)

### **Key Concepts**

At the end of this class students will understand that

- The key to configuring a desktop client application is the GetCapabilities request for the needed service
- The GetCapabilities request required by a particular client may consist of a base URL or a complete URL.
- Quantum GIS uses a base URL request model for self-configuration of WMS and WFS services
- ArcGIS uses a base URL request model for self-configuration of WMS, WCS, and WFS services
- Both Quantum GIS and ArcGIS require additional configuration to enable WFS access

### **Weekly Milestone - WMS, WFS and WCS Access in Quantum GIS**

#### **Readings**

*For reference*

- Quantum GIS [User's Manual](#), particularly the “Working with OGC Services” section.
- ArcGIS Desktop 10.1 [Overview of OGC and ISO TC211 Support](#)
- ArcGIS Desktop 10.1 [About Using OGC Service Layers](#)

## **Week 13 - Module 5.1 - Developing and Hosting OGC Services - Platforms and GeoServer Introduction**

### **Background**

Thus far we have concentrated on the client side of geospatial services oriented architectures in developing web interfaces based upon the Google Maps API, the OpenLayers javascript framework, and accessing data published using the OGC WMS, WFS, and WCS standards in desktop applications. Starting this week we begin our work on the server side - working with the GeoServer server platform to publish data through the OGC WMS, WFS, and WCS service standards. This work will demonstrate the ease with which you can share data using these standards, facilitating client use such as that that we have seen in our web site and desktop application work.

## Expected Outcomes

By the end of this class, students should be able to:

- Place files within the server file system for integration into the GeoServer platform
- Create a GeoServer *Workspace*, *Store*, and *Layer* based upon those data
- Test those layers using the *Layer Preview* tools integrated into GeoServer

## Key Concepts

By the end of this class, students should understand:

- The components of a map server platform and their relationship to each other
- The role of a geospatial server within a geospatial services oriented architecture
- The information required about data to successfully configure it for publication within GeoServer
- The stepwise process through which a dataset may be published using GeoServer

## Weekly Milestone - GeoServer Data Import

### Readings

*For Reference*

- GeoServer [Online Documentation](#): sections [Introduction](#), [Getting Started](#), and [Web Administration Interface](#)

## Week 14 - Module 5.2 - Developing and Hosting OGC Services - OGC Services and Styling in GeoServer - Part I

### Background

While integrating data into a web mapping platform is a critical first step, a robust geospatial data service system must also support the development and use of cartographically informative representations of the published data. GeoServer accomplishes this through the integration of the Open Geospatial Consortium's Styled Layer Descriptor standard into the system for publishing alternative styles for vector and raster data products. This standard allows for the configuration of attribute and scale-based representations that may be advertised by name and requested by reference through the published OGC WMS services.

## Expected Outcomes

At the end of this class students will be able to:

- Develop a basic SLD for a vector layer within GeoServer
- Define an attribute filter as part of an SLD to allow for differential representation of features based upon the values of one or more attribute values
- Define a scale limit filter as part of an SLD to control the scales at which a defined SLD rule will be executed

## Key Concepts

At the end of this class, students will understand:

- The role of the OGC SLD standard in defining and publishing geospatial data via WMS
- The capabilities of the SLD standard to defined representation styles that may be defined in reference to attribute values and WMS request scales
- That logical comparison operators may be used to control the application of style rules to specific features

## Weekly Milestone - Styling of Layers in GeoServer

### Deep Dive - Data Integration and Styling in GeoServer

## Readings

*For Reference*

- GeoServer [Online Documentation](#): sections [Working with Vector Data](#), [Working with Raster Data](#), [Working with Databases](#), [styling](#), [Filtering in GeoServer](#), and [services](#)

## Week 15 - Module 5.2 - Developing and Hosting OGC Services - OGC Services and Styling in GeoServer - Part II

### Background

Thus far we have discussed the general workflow for publishing geospatial data using GeoServer, the general concepts relating to styling those data for presentation through the web services published by GeoServer, and specific methods for styling vector data, and applying styles to specific subsets of vector data through the use of filters. This week we conclude our discussion of GeoServer with a demonstration of the methods used in the styling of raster data. Without the application of meaningful styles, raster data can be visually unintelligible. You will learn the tools that you have within the OGC Styled Layer Descriptor standard (as implemented within GeoServer) to define and apply meaningful styles to raster data - ultimately producing more useful visualizations of those data.

### Expected Outcomes

At the end of the class, students will be able to:

- Define ColorMaps for raster data that produce a continuous gradient of colors, a stepped color ramp between defined raster values, and color assignments for specific raster values.
- Map bands from a multi-band raster file to the RGB colors used to generate a color map image or to a single grayscale band for generating a single-band representation.
- Define opacity for an entire raster dataset or on a per-value level within a ColorMap.
- Apply a variety of contrast enhancements to individual color bands to improve/modify the display of the overall map image.

## Key Concepts

At the end of the class, students will understand:

- The concepts of continuous color ramps, color intervals, and colors assigned to individual raster values as alternative representation models for raster data
- The concept of a multi-band raster image and the mapping of specific raster bands into the RGB color space for visual representation
- Two methods for defining opacity for a raster dataset as a whole or for individual values
- The options for and potential effects of contrast enhancement on raster map images.

## Weekly Milestone - none

## Readings

*For Reference*

- GeoServer [Online Documentation](#): sections [Working with Vector Data](#), [Working with Raster Data](#), [Working with Databases](#), [styling](#), [Filtering in GeoServer](#), and [services](#)

## Week 16 - Module 6 - Class Review

### Background

We've now reached the end of the course where we have worked extensively within both the data (and data publication), and client tiers of a tiered geospatial services oriented architecture (SOA). In particular we've worked with GeoServer as a visualization and data publication platform and OpenLayers, Quantum GIS, and ArcGIS as consumers (clients) of the OGC services that GeoServer publishes. We've also worked with Google Maps as a proprietary web mapping platform and done hands-on work with the interrogation of the Open Geospatial Services that provide the glue between the SOA tiers. With this experience I hope that you've have gained an understanding of the value of these Internet services for both enhancing your ability to bring data and map images into your applications and share your data with others.

### Expected Outcomes

At the end of the class students will be able to:

- publish geospatial data using GeoServer using the OGC WMS, WFS, and WCS service standards
- style published data using the OGC SLD standard
- configure Quantum GIS and ArcGIS to consume OGC Services (WFS and WMS for Quantum GIS; WFS, WMS and WCS for ArcGIS)
- develop Google Maps-based mapping clients using version 3 of the Google Maps API
- develop OpenLayers-based web mapping clients (supporting WMS, WFS, and KML integration).

## Key Concepts

At the end of the class students will understand:

- the tiers of a geospatial services oriented architecture

- where components like GeoServer, ArcGIS, Quantum GIS, Google Maps, and OpenLayers fit into such an architecture
- the roles of open interoperability standards such as those from the Open Geospatial Consortium (WMS, WFS, WCS, SLD, KML), W3C (HTML, CSS) and EMCA (Javascript) in enabling connectivity between and capabilities within each tier

### **Weekly Milestone - A Final OpenLayers Mapping Client**

#### **Readings - none**

This work by Karl Benedict is licensed under a Creative Commons Attribution-ShareAlike 4.0 International License.