DOS

1949

دائرة الاحصاءات العامة

Android/GIS and Web Service

Java Introduction and Overview

- Object Oriented Programming.
- Switch , Loops, If-else, Methods, Arrays.
- Exception handling.
- File I/O and Streams.
- Java Virtual Machine.

Java Introduction and Overview

Object Oriented Programming

- Object , Class.
- Properties ,Methods, and Encapsulation.
- Polymorphism.
- Inheritance.
- Public VS Private

Java Introduction and Overview

Object Oriented Programming

- Constructor and Destructor.
- Method Overloading/Overriding.
- Interface.

Java Introduction and Overview

Switch

```
switch ( variable_to_test ) {
case value:
code_here;
break;
case value:
code_here;
break;
default:
values_not_caught_above;
}
```

Java Introduction and Overview

If - Else

- **if (** condition_to_test **) {**
  }
  else {
  }

Java Introduction and Overview

For Loops

-
  ```
  for ( start_value; end_value; increment_number ) {
   //YOUR_CODE_HERE
   }
  ```

Java Introduction and Overview

For each

- For(item:items)
- { //code here
- }

Java Introduction and Overview

While Loops

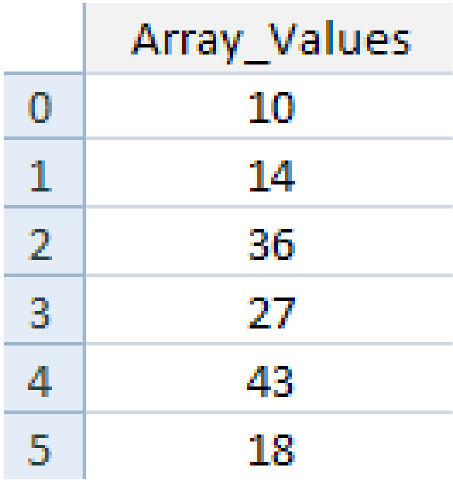
- **while (** condition **) {**
  // Your code here

  }

Java Introduction and Overview

Arrays

- Array is a way to hold more than one value at a time.
- Example: **int[ ] aryNums = new int[6];**



	Array_Values
0	10
1	14
2	36
3	27
4	43
5	18

Java Introduction and Overview

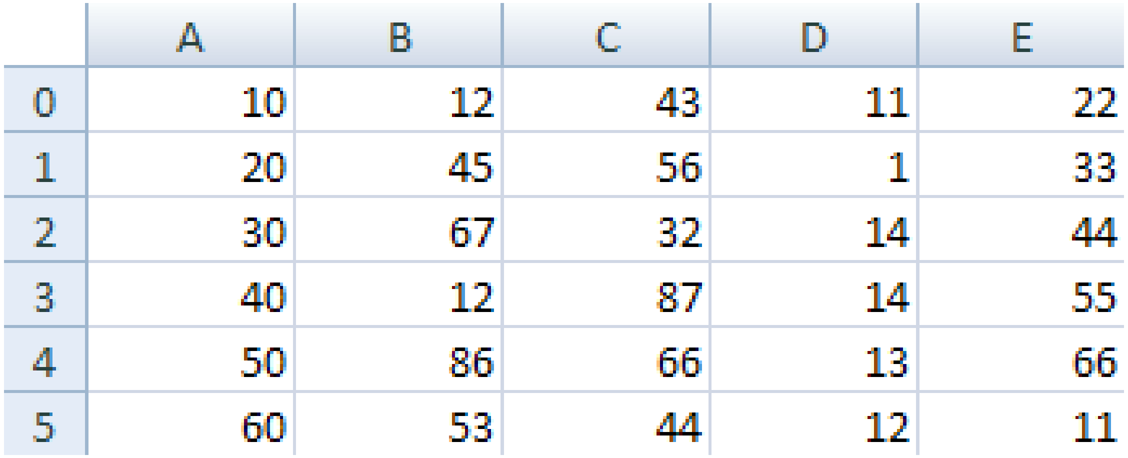
Sorting Arrays

- Java.util.Arrays;
- **Arrays.sort(yourArrayName);**

Java Introduction and Overview

Multi Dimensional Arrays

- **int[ ][ ] aryNumbers = new int[6][5];**

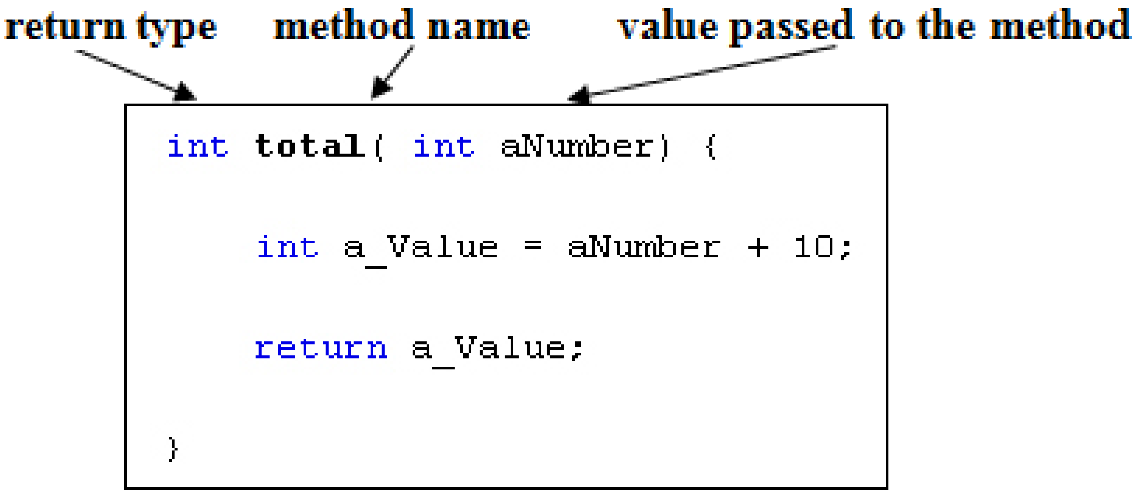


	A	B	C	D	E
0	10	12	43	11	22
1	20	45	56	1	33
2	30	67	32	14	44
3	40	12	87	14	55
4	50	86	66	13	66
5	60	53	44	12	11

Java Introduction and Overview

Methods

- **Method Declaration.**
- **Method Calling.**



```
int total( int aNumber) {

    int a_Value = aNumber + 10;

    return a_Value;

}
```

Java Introduction and Overview

Exception Handling

```
try { //Protected code }
catch(ExceptionType1 e1)
{ //Catch block }
catch(ExceptionType2 e2)
 { //Catch block }
catch(ExceptionType3 e3)
 { //Catch block }
finally { //The finally block always executes. }
```

Java Introduction and Overview

File I/O and Streams

- Stream : a sequence of data.
- Input Stream : read data from a source .
- Output Stream : write data to a destination .
- Some of most used streams classes :

- Byte Streams.

- Character Streams.

Java Introduction and Overview

Byte Streams

- **Byte Streams** : are used to perform input and output of 8-bit bytes.
- Used classes are :

  - **FileInputStream** .

  - **FileOutputStream**.

Java Introduction and Overview

Character Streams

- **Character Streams :** are used to perform input and output for 16-bit Unicode(2 bytes at a time).
- Same as Byte Stream but these classes reads and writes 2 Bytes at a time.
- Used classes are:

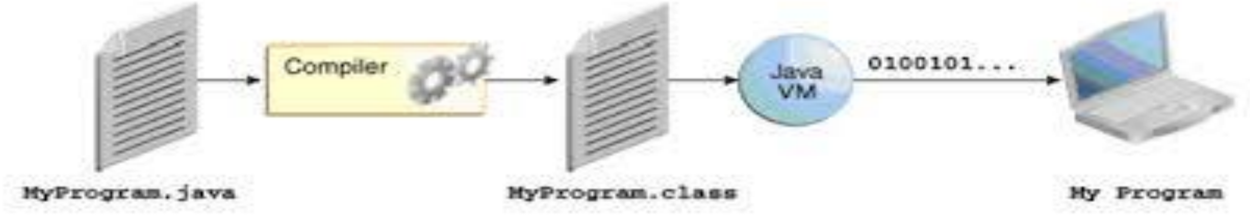
  -FileReader .

  -FileWriter

Java Introduction and Overview

Java Virtual Machine

- An implementation of the Java Virtual Machine Specification, interprets compiled Java binary code (called byte code) for a computer's processor (or "hardware platform") so that it can perform a Java program's instructions.



What is Web Service?

- Many organizations use multiple software systems for management. Different software systems often need to exchange data with each other, and a Web service is a method of communication that allows two software systems to exchange this data over the internet. The software system that requests data is called a *service requester*, whereas the software system that would process the request and provide the data is called a *service provider*.

Java Web Service Introduction

- Client – Server concept:

  - Create (deploy)

  - Consume(call).

- SOAP(Simple Object Access Protocol).
- WSDL(Web Service Description Language).
- SOAP web service.
- RESTful (Representational State Transfer)web service.

Java Web Service Introduction

SOAP(Big) Web Service

- **SOAP** provides a way to communicate between applications running on different ***operating systems***, with different ***technologies*** and ***programming languages***.

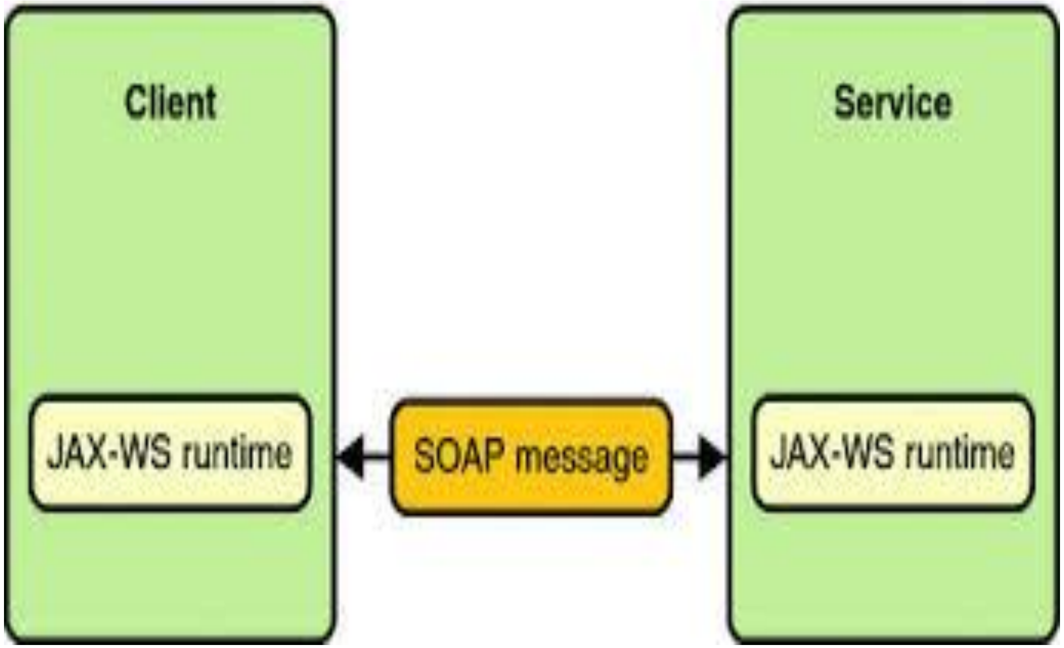
Java Web Service Introduction

SOAP(Big) Web Service

- SOAP stands for Simple Object Access Protocol.
- SOAP is a protocol for accessing web services.
- SOAP is based on XML.

Java Web Service Introduction

SOAP(Big) Web Service



Java Web Service Introduction

SOAP(Big) Web Service

- JAX-WS: ***J***ava ***A***PI for ***X***ML ***W***eb ***S***ervice.
- WSDL based .
- Used for: Professional enterprise application integration scenarios with a longer lifespan.
- XML used for data send and receive(messages).

Java Web Service Introduction

WSDL for SOAP Web Service

- WSDL : WSDL stands for Web Service Description Language , it is an XML file that describes the technical details of how to implement a web service.

- Using this WSDL file we can understand things like:

  - Port / Endpoint – URL of the web service.
  - Input / Output message formats.
  - Security protocol that needs to be followed .
  - Which protocol the web service uses.

Java Web Service Introduction

RESTful Web Service

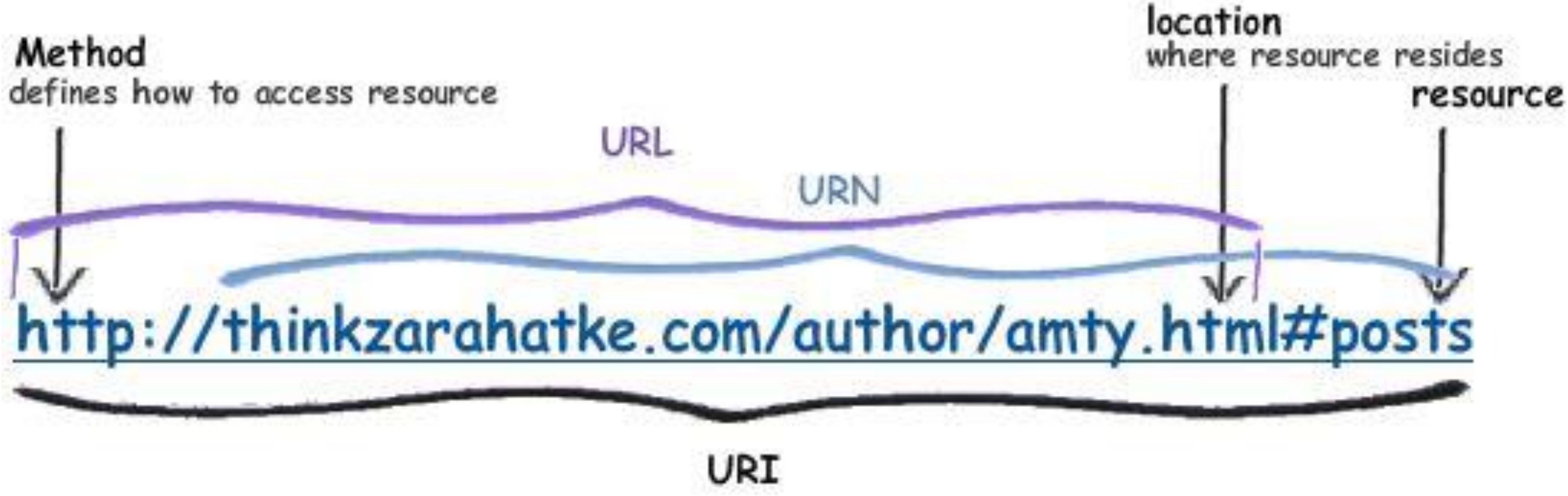
- JAX-RS : ***J***ava ***A***PI for ***R***ESTful ***W***eb ***S***ervice.
- When and Why to use RESTful web service:

  - PDAs and mobile phones(limited bandwidth).

  - Lightweight .

  - Stateless Resources.
- Ex. : Flickr, Google Maps and Amazon .

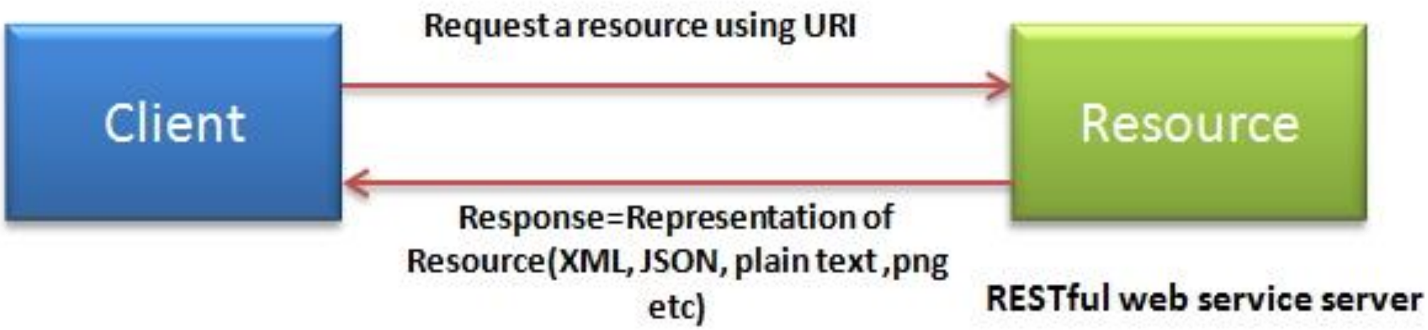
URI/URL/URN


Method
defines how to access resource
location
where resource resides
resource
URL
URN
http://thinkzarahatke.com/author/amty.html#posts
URI

Ex: http://www.w3.org/Icons/w3c_home

Java Web Service Introduction

RESTFUL Web Service


Request a resource using URI
Client
Resource
Response=Representation of Resource(XML, JSON, plain text ,png etc)
RESTful web service server

Java Web Service Introduction

Send - Receive Data in RESTful WS

- RESTful web services use HTTP protocol methods for the operations they perform are: GET (Read), POST(Create) ,
  PUT (Update), and DELETE (Delete).
- Some types of transferred data are :
  XML , JSON,etc….

Java Web Service Introduction

XML for RESTFUL WS

Example :

```
<employees>
        <employee>
            <firstName>John</firstName><lastName>Doe</lastName>
        </employee>

        <employee>
            <firstName>Anna</firstName><lastName>Smith</lastName>
        </employee>

        <employee>
            <firstName>Peter</firstName> <lastName>Jones</lastName>
        </employee>
    </employees>
```

Java Web Service Introduction

JSON for RESTFUL Web Service

- JSON: **J**ava**S**cript **O**bject **N**otation.
- JSON is a syntax for exchanging data.
- **JSON** is an easier-to-use alternative to **XML**.
- can be read and used as a data format by any programming language.

Java Web Service Introduction

JSON for RESTFUL Web Service

Example:

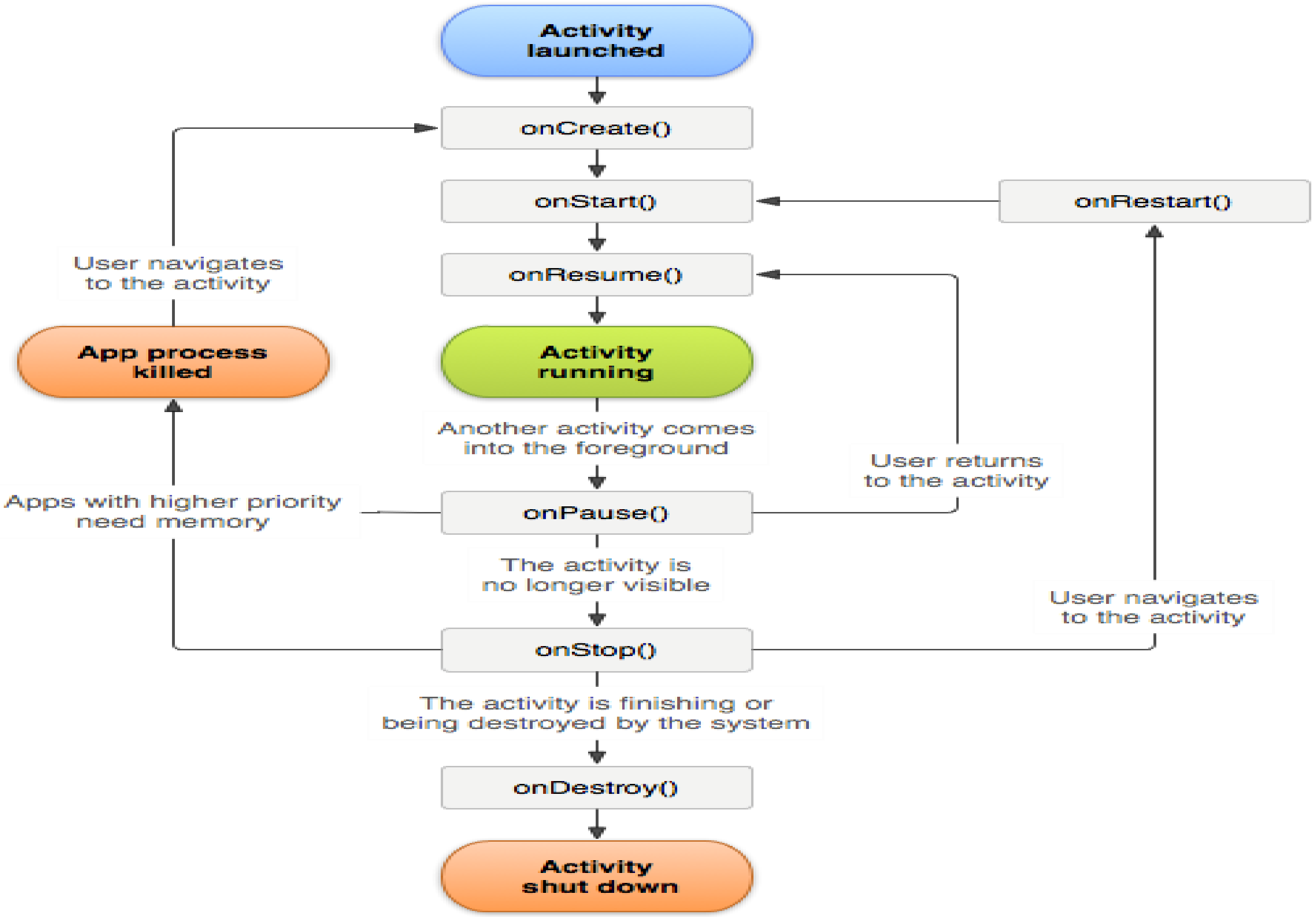
```
{"employees":[
    {"firstName":"John", "lastName":"Doe"},
    {"firstName":"Anna", "lastName":"Smith"},
    {"firstName":"Peter", "lastName":"Jones"}
  ]
}
```

Android Studio and Android Introduction

- Eclipse .
- Android SDK .
- Layout XML file.
- Activity : onCreate(),onPause().
- AndroidManifest.xml
- Android Virtual Device.
- **A**ndroid **P**ac**K**age files (APK).

Android Studio and Android Introduction

Activity Life Cycle



Activity launched
onCreate()
onStart()
onRestart()
onResume()
User navigates to the activity
App process killed
Activity running
Another activity comes into the foreground
User returns to the activity
Apps with higher priority need memory
onPause()
The activity is no longer visible
User navigates to the activity
onStop()
The activity is finishing or being destroyed by the system
onDestroy()
Activity shut down

Hello World Sample Application

Hello World

☺

Connecting android application with webservice and synchronizing mobile data with desktop

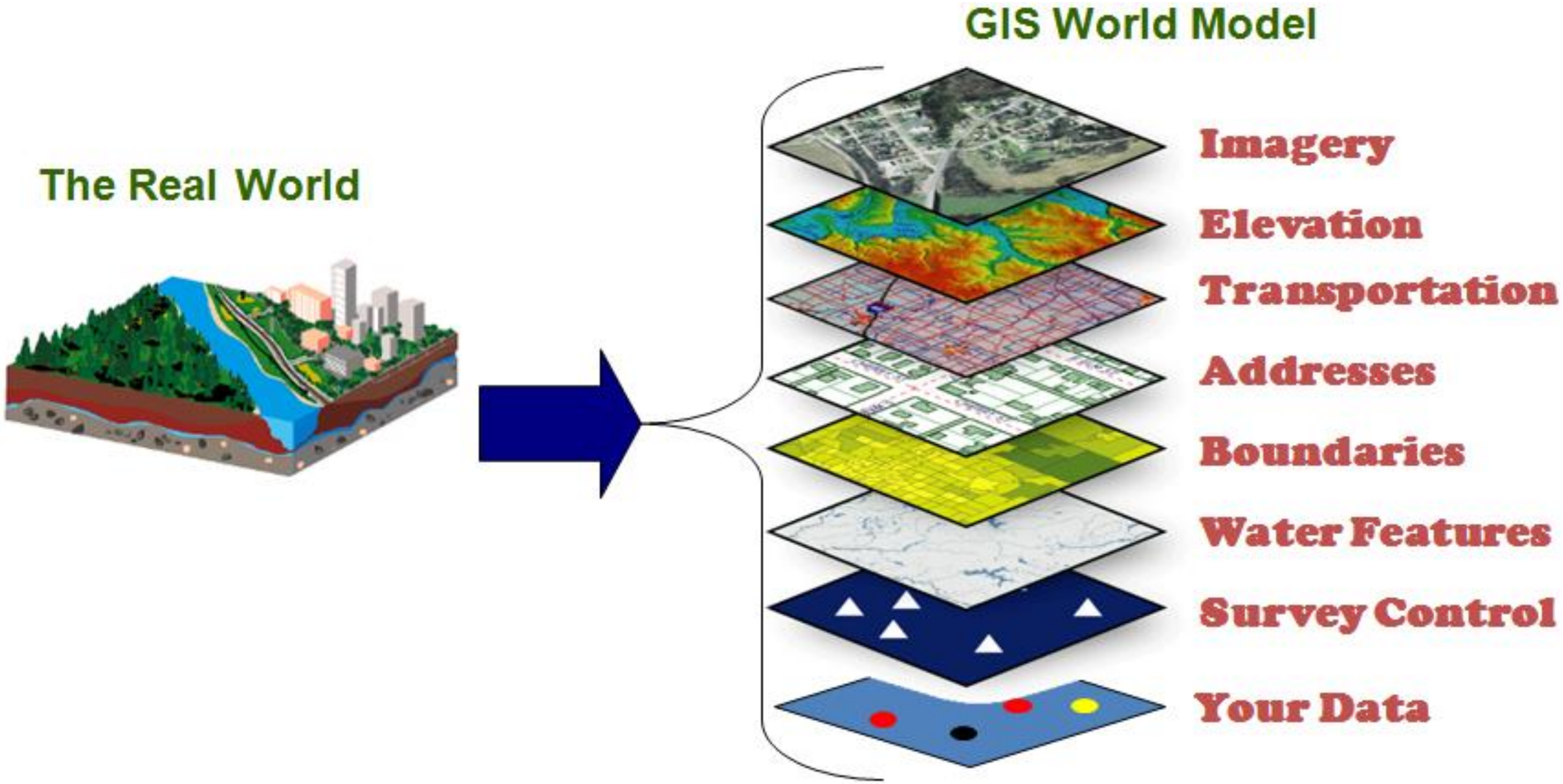
GIS Introduction, Maps, and Layers

- Desktop , Server, Mobile, Development.
- ArcGIS (ESRI).
- ArcMap.
- ArcGIS web map.
- Sharing Maps (online , offline)
- Open Source GIS .

GIS Introduction, Maps, and Layers

- Vector : points, Lines, Polygons
- Mandatory Files : .shp , .shx , .dbf .
- Other Files : .prj , .sbn, .mxs , .qix ...others
- Shape file must not contain both polyline and polygon data.
- Shape file: collection of files with a common file prefix stored in the same directory.
- .shp file needs other supporting files to work.

GIS Layers

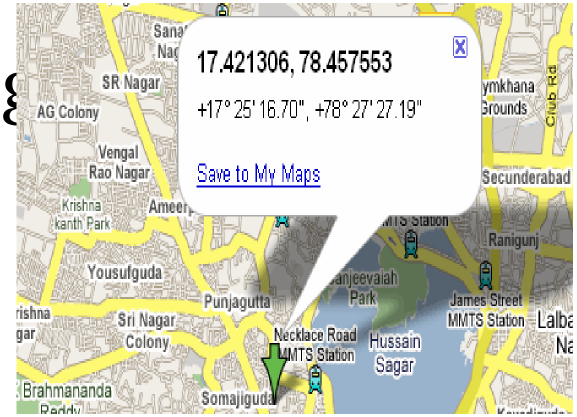

GIS World Model
The Real World
Imagery
Elevation
Transportation
Addresses
Boundaries
Water Features
Survey Control
Your Data

Maps partitioning, storing, and assigning

- Covered above ?

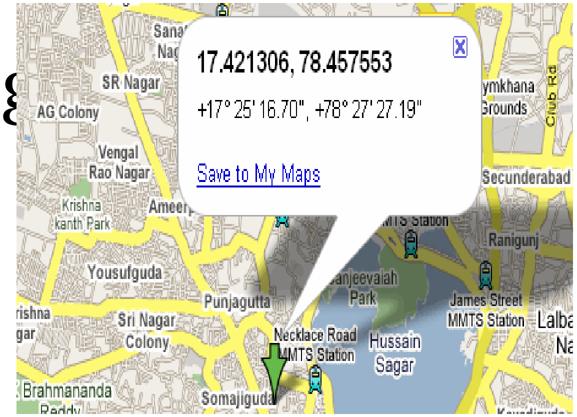
Loading Maps on Tablet and Working with GPS Coordinates

- *android. Location* API: determine the current geolocation.
- To obtain the user location in your application, you can use both GPS and the Network Location Provider.
- Geocoding and Reverse Geocoding
- latitude and longitude.



Loading Maps on Tablet and Working with GPS Coordinates

- *android. Location* API: determine the current geolocation.
- To obtain the user location in your application, you can use both GPS and the Network Location Provider.
- Geocoding and Reverse Geocoding
- latitude and longitude.



Loading Maps on Tablet and Working with GPS Coordinates

- Activate GPS on your test device or Emulator.
- Activating GPS on Emulator in one of two ways:

- The Google Map *activity* should automatically activate the GPS device in the emulator.

- Start Google Maps on the emulator and request the current geo-position, this will allow you to activate the GPS.

http://www.vogella.com/tutorials/AndroidLocationAPI/article.html

Geoposition in Android

- Add permissions to *AndroidManifest.xml file:*

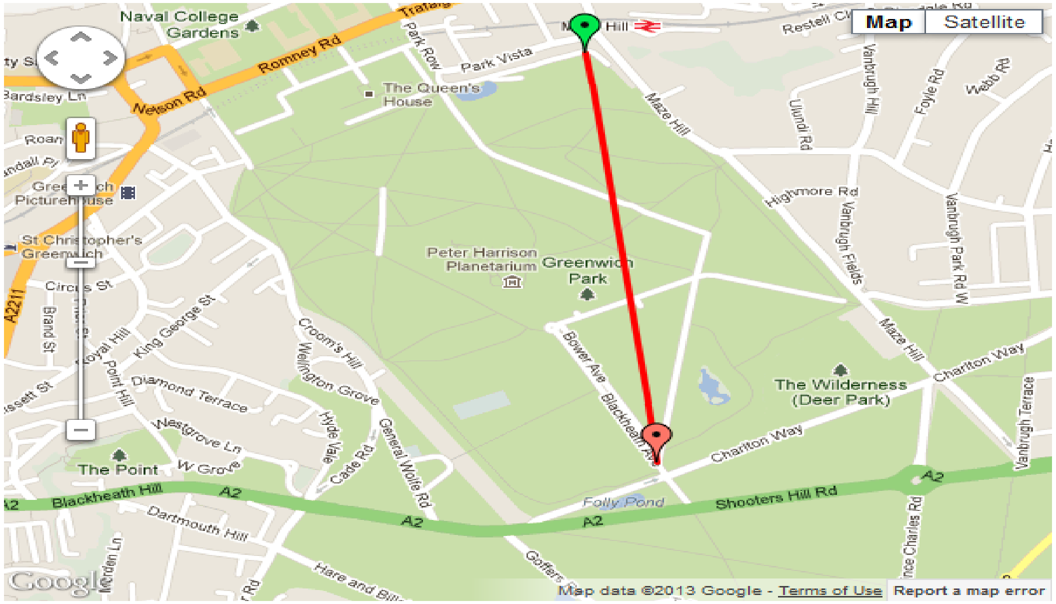
- *INTERNET*

- *ACCESS_FINE_ LOCATION*

- *ACCESS_COARSE_LOCATION*

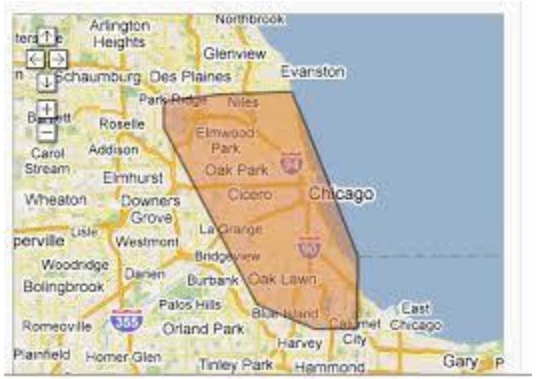
Manipulating Points and Polylines

- polyline.getPoints().
- polyline.setPoints().
- polyline.remove().



Manipulating Polygons

- Polygon auto completion.



Synchronizing Data with Server

- Synchronizing data with server

Prepared and Presented by

Nabeel Adnan

nabeelnar@gmail.com

Thank You