Chapter 1

Date: 2075-8-06. Page:

Enterprise

Enterprise essentially means that the company has multiple levels, locations divisions or departments that all collaborate to achieve big picture business objectives.
For eg customer relationship management is often refereed as enterprise application [large scale business]

Enterprise application

It is an application that a business uses to support the organization in order to solve enterprise problem.
It denotes a software platform that is quiet complex and comparatively huge for small business use.
Enterprise application demands high requirements in terms of security, user access and maintainance
[application for large scale business]

Date: Page:

Imp # Technologies used to develop Enterprise application

User Interface Layer (Presentation Layer)	AWT, Scoring, HTML, JSP, Velocity Framework
Business Processing Layer (Business Layer)	Servlets, JSP, EJB
Data storage and Access layer (Persistance layer)	JDBC, EJB

⇒ Not imp for exam

Java →
- J2SE/JSE/Java SE (Standard edition (Introduction, OOP, String manipulation ----
- J2EE/JEE/Java EE Servlets, JSP, EJB, middleware service
- J2ME/JME/Java ME (MIDP Wireless messaging, Mobile XML + JSE EA

GURUKUL

Date: Page:

Java is widely used for developing EA Because:

(i) Learning curve

(ii) Java has a vast array of Libraries, frameworks, tools and IDEs and server provider

(iii) Platform independence

(iv) Maturity

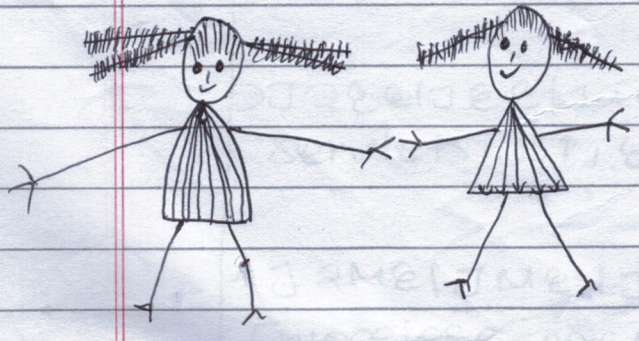
(v) Java ensures security, stability, robuteness, high perpermance etc.

2075-8-10

Date: Page:

* Example of Enterprise Application:
  - → Customer Relationship management-CCR
  - → Automated billing system
  - → enterprise resources planning
  - → HR management.

* Framework:
  - A framework is a real or conceptual structure intended to ~~solve~~ a serve as a support or guide for the buildin of something (that expands the struc into something useful.)
  - In computer systems, a framework is often a layered structure indicatin what kind of programs, can or shoul built and how they would interact



Imp ✓ Date: Page:

* Enterprise Application VS Desktop Application Vs web application.

a) Desktop Application:

(1) → runs on a single tier architecture.

(2) → All the presentation logic, business logic and data storage and access logic resides in a single machine.

(3) → Moreover it can be part of network but the application and its database stay in the same machine.

1) → It doesn't support multiple users concurrently.

b) Enterprise Application:

1) → It normally includes more than a single tier.

2) → Multiple concurrent users

3) → It seems to be similar to web application but not, ~~it is superior than we~~

4) It is superior than web application.

5) → It is Java JEE compatible.

6) It have client tier (java client, browser based, mobile based client, presentation tier (JSP), business logic tier (webservices - SOAP (restful), security, transaction, Data tier (DBMS)

#tomcat : like Apache
- JSP requires webserver ie tomcat.

Date: Page:

3) Web Application

✓ It can be implemented without such complex enterprise technology

→ web application runs on web server like tomcat while for enterprise application, you need an application server like glassfish server.

→

GURUKUL

Date: Page:

2075-8-12

EJB (Entity Java Bean)

1) → It is used to develop scalable, robust & secured enterprise application using java.

2) → It is a server-side component that encapsulates the business logic of an application.

3) → The business logic is the ~~code~~ that ~~fullfills~~ fulfills the purpose of the application.
For eg: In an inventory control application, the enterprise beans might implement the business logic in methods called checkInventoryLevel and orderProduct. By invoking these methods, clients can access the inventory services provided by the application.

* Benefits of EJB

1) - EJB simplify the development of large distributed application.

2) - EJB container provides system ~~level sex~~ level services to enterprise beans.

3) - It is responsible for system level services, such as transaction management & security authorization.

4) - Enterprise beans are portable components.

GURUKUL

Date: Page:

1

* Types of enterprise Beans.

classification	Subclassification.
1) Session Bean	1a) Stateless session Bean b) Stateful " " c) Singleton " "
2) Entity Bean	2a) BMP(Bean Managed Persistence 2b) CMP(Container Managed Persistence)
3) Messagedriven bean	None

Web Technologies:

* Protocol:

- It defines standard for enabling the connection communication and data transfer between a places on a network.
- It is a system of rules for message exchange among computers.

→ General requirement for a protocol:

1) data formats for data exchage.

header	data

2) Address formats for data exchage
   - Sender & receiver
   - address are stored in header section.

Date: Page:

3) Address Mapping
4) Routing
5) Detection of transmission errors.
6) Acknowledgement
7) Loss of information (handle)
8) Sequence control
9) Flow control.

Stateless protocol
- HTTP
- UDP

Stateful protocol
- TCP
- FTP

* HTTP request methods.

1. POST
2. GET
3. HEAD
4. PUT
5. DELETE
6. connect
7. OPTIONS
8. PATCH
9. TRACE

1. POST → used to submit an entity to the specified resources
2. GET → request a representation of a specified resources.
3. HEAD → ask for a respons as get method but without body response.
4. PUT - replace all current representation of target resources.

Date: Page:

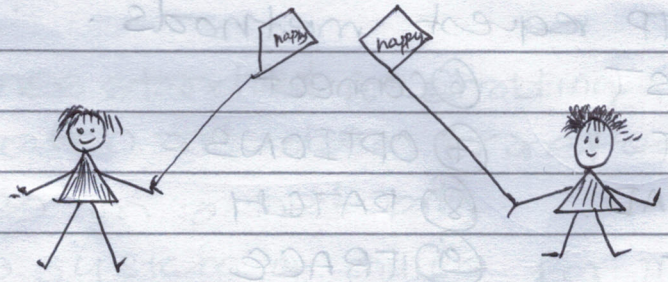
5) DELETE → delete specified resources.

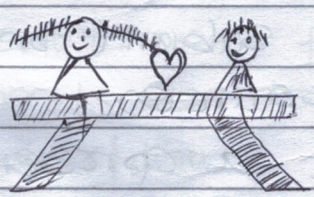
6) CONNECT → establishes a tunnel to the server identified by resources.

7) OPTIONS → to describe communication options.

8) PATCH → is used to apply partial modification of resources.

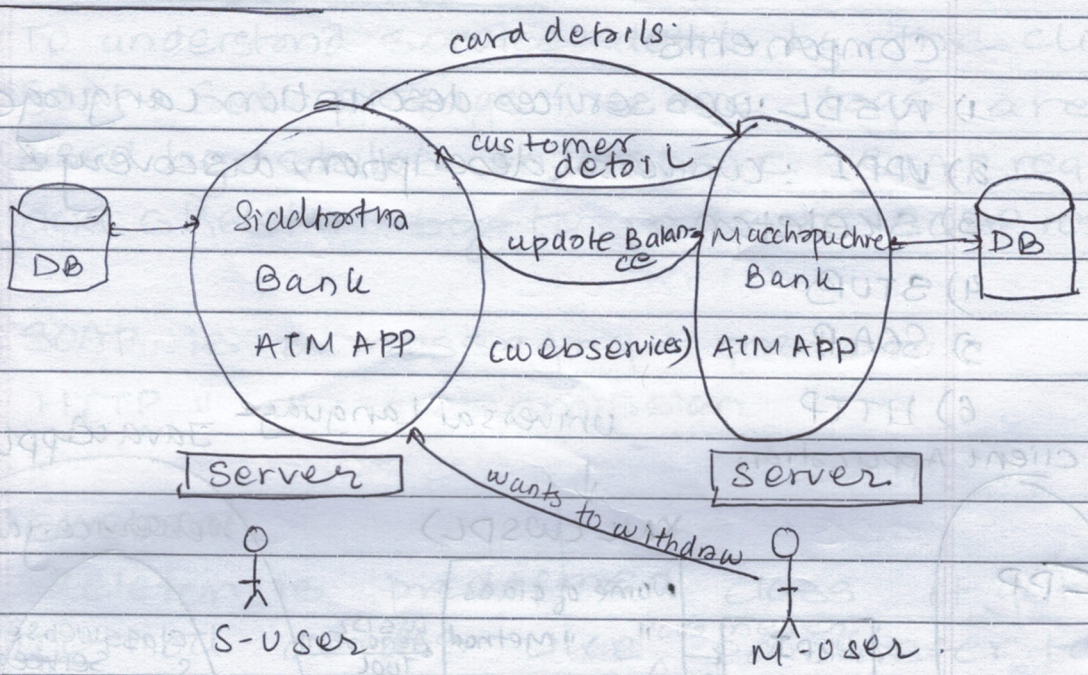
9) TRACE → performs a message loop back test along the path to target resources.





2075-8-13 Date: Page:

Web services



Siddhartha Bank never provide direct access to its database to other system ~~so as~~ or a another bank for security purpose.

Also one bank application can use one programming language while another can use another programming language.

Benefits

1 (i) Facilates Reuse of existing asseb
(ii) Lower cost of maintainece
(iii) Interopeability in heterogeneous environment
(iv) Freedom of choice
Programming productivity.

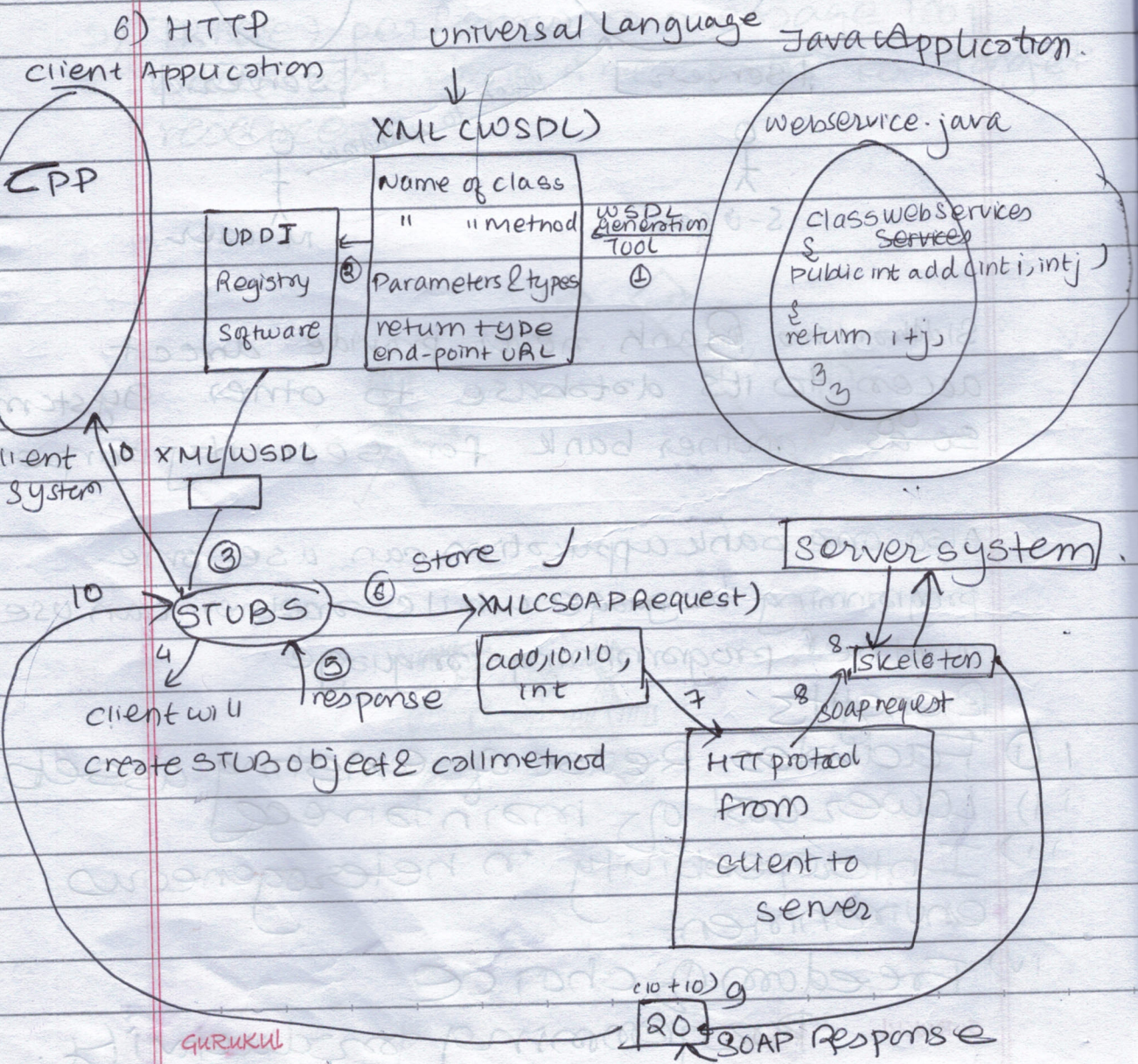
GURUKUL

Date: Page:

Web Service Architecture.

Components:

1) WSDL: web services description language.
2) UDDI : universal description, discovery & integration
3) skeleton
4) STUB
5) SOAP
6) HTTP



GURUKUL

Date: Page:

Purpose of ~~WSDL~~ WSDL:

WSDL → To understand service detail by the client.
SOAP → Set of tags, these tags are
STUBS → used by stubs to prepare SOAP request
SOAP → and skeleton ~~by~~ to prepare SOAP response

SOAP is a messaging protocol
HTTP " " transmission "

* SKELOTON
Skeleton is predefined class it gets method details like parameter, parameter types, return, types etc from SOAP request, and (It will invoke that client requested method on (actual service class) and (get return value) and is stored in soap response.

* STUB
. It receives method called from client.
It would store the method details into SOAP request and receive, returned value from SOAP response and hand over to client application.

* ~~How communication~~ How communication betn
GURUKUL 2 diff application occurs
(Inter operabilities).

Date: Page:

~~How communication between ?~~

Data exchange Formates.

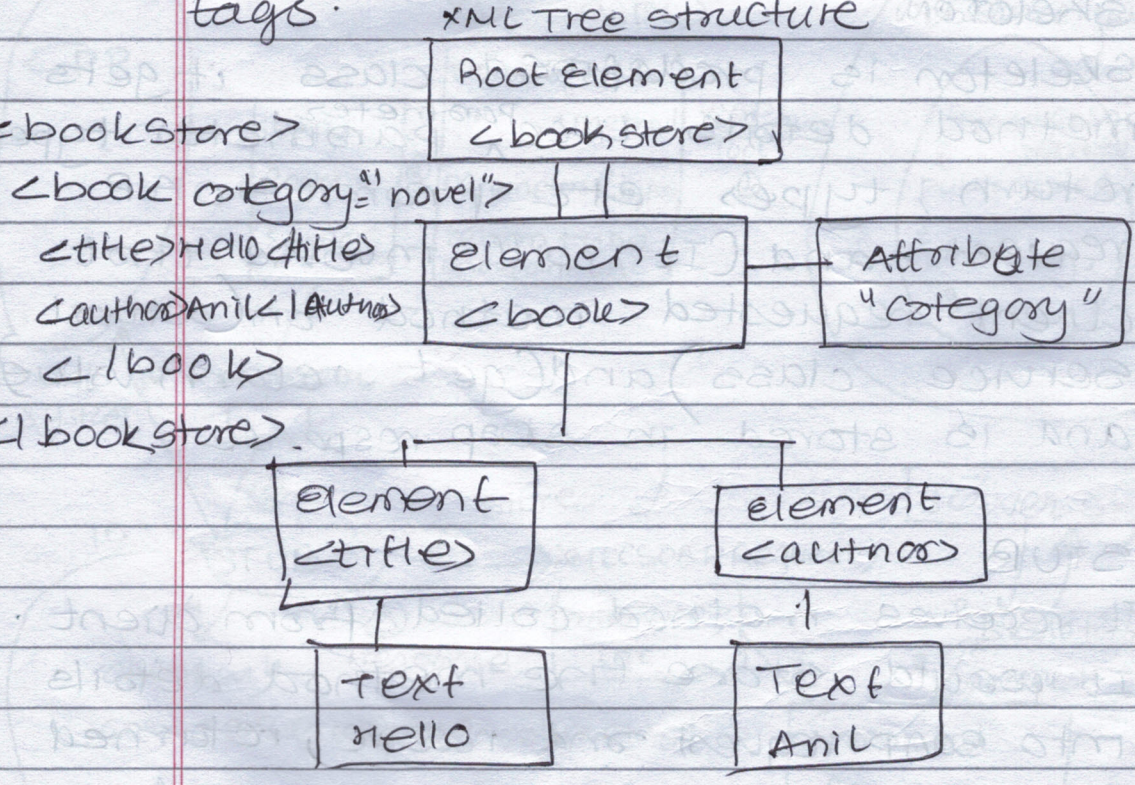
① XML:

→ extensible markup language.

→ XML doesnot do anything. It is just a info wrapped in tags.

→ XML was designed to carray data (HTML is to display data).

→ XML tags are not predefined like HTML tags.



Date: Page:

Rules:

1) All XML elements must have a closing tag
2) XML tags are case sensitive.
3) XML elements must be properly nested.
4) XML Attribute values must be quoted.

- Example.

* XML namespace	* XML Prolog
Syntax xmlns:prefix="URI"	<?xml version="1.0"
eg	encoding="UTF-8" ?>
<root>	this is xml

```
<h:table xmlns:h="http://www.w3c.org/TR/html4/">
  <h:tr>
    <h:td>Apple</h:td>
  </h:tr>
</h:table>
<f:table xmlns:f="...">
  <f:name>coffeeTable</f:name>
</f:tables>
</root>
```

GURUKUL

Date: Page:

① JSON

syntax
"name" : "John" ← must be double quotes
↑ require for JSON and no for Javascript

example JSON.

Emp

```
{"employees":[
  {"firstname":"John", "LastName":"Doe"},
  {"firstname":"Anna", "LastName":"Smith"}
]}
```

XML

eg.

```
<employees>
  <employee>
    <firstname>John</firstname>
    <lastname>Doe</LastName>
  </employee>
  <employee>
    <firstname>Anna</firstname>
    <lastname>Smith</lastname>
  </employee>
</employees>
```

Date: Page:

```
<?xml version="1.0" encoding="UTF-8"?>
<order id="1234" date="05/06/2013">
  <customer first-name="James"
            last-name="Rorrison">
    <email>james@me.com</email>
    <phoneNumber>+44 1234 12342</phoneNumber>
  </customer>
  <content>
    <order-line item="H2G2" quantity="1">
      <unit-price>28.5</unit-price>
    </order-line>
    <order-line item="Harry Potter" quantity="2">
      <unit-price>34.99</unit-price>
    </order-line>
  </content>

  <credit-card number="1357" expiry-
               date="10/13" control-number="234
               "type="visa" />
</order>
```

GURUKUL

```
{"?xml": {"version": "1.0",
          "@encoding": "UTF-8"
         },
"order": {"@id": "234"
          "@date": "05106/20B",
          "customer":
           {"@first-name": "James",
            "@last-name": "Morrison",
            "email": "   @gmail.com",
            "phone": "...",
           },
 "content": {"orderline": [
             {"@item": "H2G2",
              "@quantity": "1",
              "unit-price": "23.5"
             },
             {"@item": "Harry Potter",
              "@quantity": "2",
              "unit-price": "0..  "
             }]
 },
 "credit_card": {"@numbes": "1357",
                 "@exp_date": "10/13",
                 "@card_numbes": "234",
                 "@type": "visa"
                }
}
```

Date:_____ Page:_____

```
{"country":[
{"name":"Nepal","capital":"Kathmanda",
 "feature":"everest"},
{"name":"India","capital":"NewDelhi",
 "feature":"Taj-mahal"},
{"name":"china","capital":"Beijing",
 "feature":"Great wall"}]}
```

```
<country>
  <name>Nepal</name>
  <capital>Kathmandu</capital>
  <feature>Taj-mahal</feature>
</country>
<country>
  <name>India</name>
  <capital>NewDelhi</capital>
  <feature>Taj-mahal</feature>
</country>
<country>
  <name>china</name>
  <capital>Beijing</capital>
  <feature>Great-wall</feature>
</country>
```

GURUKUL

Date: Page:

Enterprise Architecture Frameworks (EAF)

1) - Frameworks help people to organize and assess completeness of integrated models of their enterprises.

2) - An Architectural Framework gives a skeletal structure that defines suggested architectural artifacts, describes how those artifacts are related to each other and provides generic definition for what those artifacts might look like. EAF says

* how to create and use an enterprise architecture.
* Principles and practices for creating and using ~~of this~~ architectural description of the system.

Purpose of Frameworks

1) Organize integrated models of an enterprise.
2) Assess completeness of the descriptive representation of an enterprise.
3) Understand an organization of a system.
4) Assist in identification and categorization

Date: Page:

5) help to manage complexity
6) Identify the flow of money in the enterprise
7) Provide a communication mechanism.

Zachman enterprise Architecture Framework.

*	1) What (data)	2) How (function)	3) Where (network)	4) Who (people)	5) When (time)	6) Why (motivation)
Scope ontext						
Business model						
System model ogical)						
Technology odel physical)						
Detailed epresentation mponent)						
) Real ystem executing ne game of aseball						

Fig: An Empty zachman Framework

GURUKUL

Date: Page:

(a) - The Zachman Framework is an EAF which provides a formal and highly structured way of viewing and defining an enterprise.

(b) → ZF is an scheme for organizing architectural artifacts (in other words, design documents, specifications, models) that takes into account both, whom the artifacts targets (for eg business owner & builder) and what particular issue (for eg data and functionality) is being addressed.

(c) The rows present:
- different persepectives of the enterprise
- Different view of the enterprise
- diff roles in the enterprise.

About ZEA

1) Scope describes the system's vision, mission, boundaries, architecture and constraints. The scope states what the system is to do. It is called a black box model, because we see the inputs and output, but the not the inner workings.

GURUKUL

Date: Page:

2) Business model shows goals, strategies and processes that are used to support the mission of the organization.

3) System model contains system requirements objects, activities and function that implement the business model.

4) Tech model consider the constraints of human, tools, technology and materials.

5) Detailed representation, presents individual, independent components. that can be allocated to contractors for implementation.

6) Real System depicts the operational system under consideration.

GURUKUL

Date: Page:

Government Enterprise Architecture Frameworks.

1) → GEA framework promotes consistency in the ~~way~~ government models & its business process, service & infrastructu

2) GEA guides our transfromation tocoards coherent digitol goverment.
It includes
- a layer architecture framework
- Reference architectures
- toolkits and guidelines
- GEA standard reference

3) It guide on organizations transitionto working in a connected and digitol world. It
- provides a unifying common language
- categories and describes shared product and services - common capabilities
- ~~categories~~ describes and ~~defines~~ relevant standards and guidelines
- defines set of reference models and patten
- defines and identifies government business capabilites

GURUKUL

Date: Page:

4) Benefits

1) ~~und~~ Helps in to understand your current business

2) identify opportunities for change

3) model your business goals and the capabilities and investments you need to deliver them

4) design a future operating model that is technically and strategically aligned with all-of-government programmes and services

Architecture Resources

1) Governance & maturity tools

2) Information discovery process guidelines and template

3) Information assets catalogue template

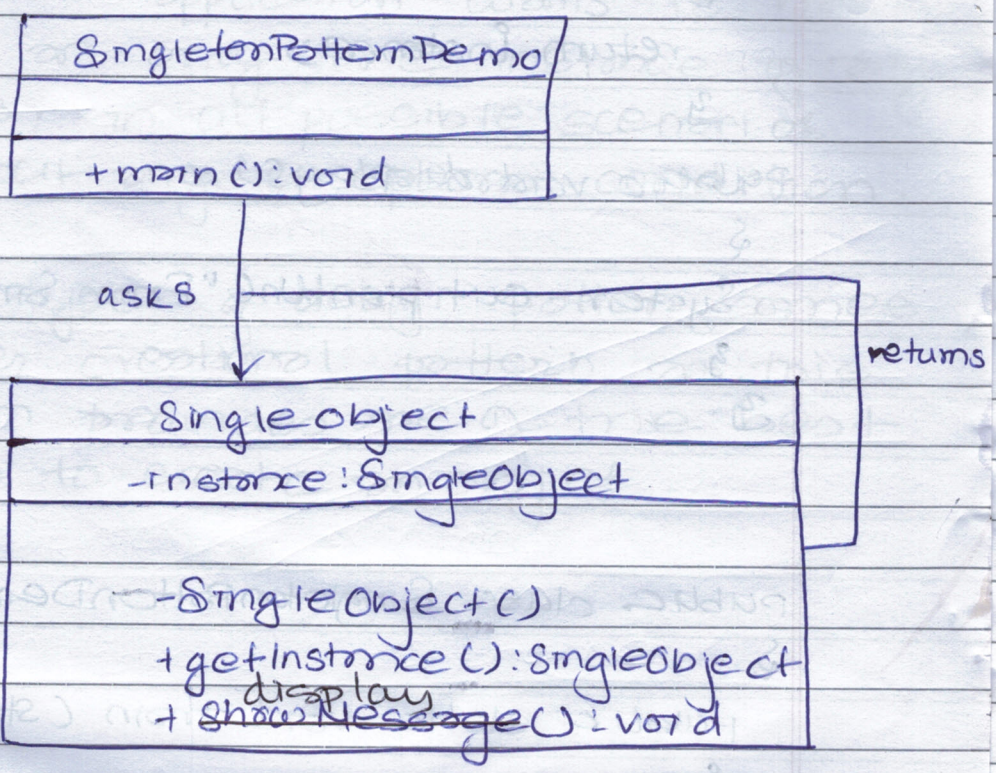
GURUKUL

Chapter - 3

Date: Page:

Design Pattern

Singleton




```
public class SingleObject {
   //create an object of SingleObject

   private static SingleObject instance =
                  new SingleObject();

//make the constructor private so that this
class cannot be instantiated
   private SingleObject() {
}
```

Date: Page:

```
public static SingleObject getInstance()
  {
    return instance;
  }
public void display()
  {
  System.out.println("From SingleObject class");
  }
}
```

```
public class SingletonPattenDemo
{
  public static void main (String[] args)
  {
    SingleObject.getInstance();
    SingleObject so = SingleObject.getInstance
    so.display();
  }
}
```

Output:

From SingleObject class

Date: Page:

1) Singleton pattern is a design solution where an application wants to have one and only one instance of any class, in all possible scenarios without any exceptional condition.

- This type of design pattern comes under creational pattern as this pattern provides one of the best ways to create an object

Date: Page:

1 Factory Pattern.

Factory pattern is one of the most used design pattern. These type of design pattern comes under creational pattern, As this pattern provides one of the best way to create object.

2 In factory pattern we create object without exposing the creation logic to client and refered to newly created object using a common interface.

For eg

Car factory

Interface

Car.java

```
Interface Car
{
  public void cost();
}
```

Date: Page:

// concrete classes

```
class Suzuki implemets Car
  {
    public void cost()
      {
        System.out.println("The suzuki : $88");
      }
  }

class Maruti Implements Car
    {
      public void cost()
        {
          System.out.println("The Maruti : $99");
        }
    }
```

Date: Page:

The java factory class

```
class CarFactory
{
    public static getCar(String criteria)
    {
        if(criteria.equals("small"))
            return new Suzuki();
        else if(criteria.equals("big"))
            return new Maruti();
        return null;
    }
}
```

Date: Page:

```
/**
* A "driver" program to demonstrate our
* "Car Factory".
* @author Sujana
*/

public class JavaFactoryPattern example
{
  public static void main (String[] args)
    {
      //create a small car
      Car car = CarFactory.getCar("small");
      car.cost();
      //create a big car
        car = CarFactory.getCar("big");
        car.cost();
    }
}
```

Date: Page:

Lazy initialization

1) Lazy initialization is a technique where one postpones the instantiation of a object until it's first use.

2) In other words the instance of a class is created when it's required to be used for the first time.

3) Lazy initialization is a technique for a performance optimization.
It is used when data is deemed to be 'expensive' for some reason

4) One typical usage of LI is in the creation of singleton class

MVC Pattern

GURUKUL

Date: Page:

For eg

```
public class Lazy{
  private Lazy lazy = null;

  private Lazy()
   {
   }

  public static Lazy getInstance()
   {
    If (Lazy == null)
      {
        Lazy = new Lazy();
      }
      return Lazy;
    }
}
```

Also

Date: Page:

Class finalLazy

public ~~final~~ class Lazy {

private static volatile Lazy instance = null;

private Lazy()
{
}

public static Lazy getInstance()
{
if(instance == null)
{
synchronized(Lazy.class)
{ instance = new Lazy();
}
}
return instance;
}
}

Date: Page:

# public static Lazy getInstance()
{
if (instance == null)
synchronized (Lazy.class)
{
if (instance == null)
{
instance = new Lazy();
}
}
}
return instance
}
}

MVC

Chapter-2 Date: Page:

Development Process Management:

Source code management:

1) Version Control system (eg git)

A) A VCS is a tool for managing a collection of a program code that provides you with 3 imp capabilities.

a) Reversibility.
b) Concurrency
c) Annotation.

It is synonymously also known as Source code Management.

B) VSC is the combination of technologies and practices for tracking and controlling process to a project files in particular to source code, documentation and webpages.

VCS is universal as it helps with almost all aspect of a project ie inter developer communication, release mgmt, bug mgmt, etc.

II Types of VCS:

a) Centralized VCS (Traditional)
b) Distributed VCS

Date: Page:

eg, SVN

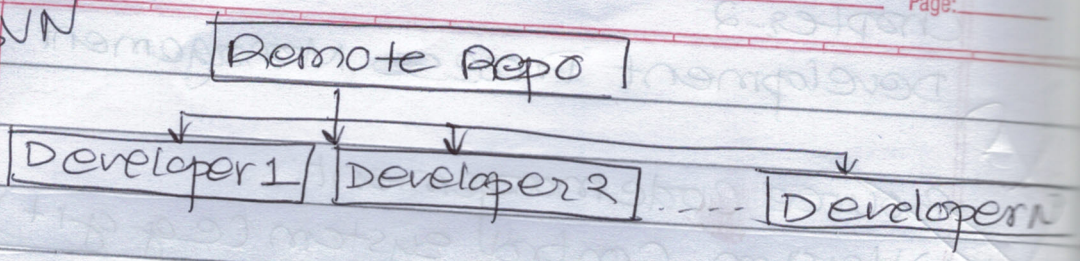


fig: centralized VCS

eg GIT

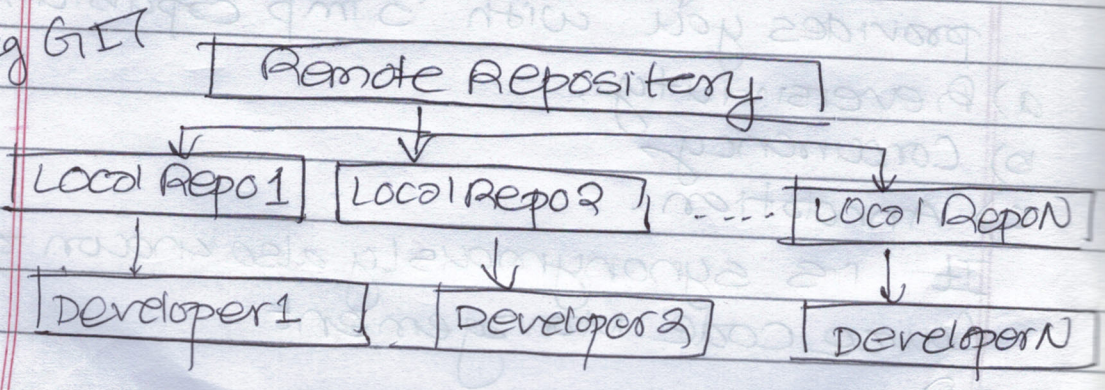


fig: Distributed VCS.

Benefits of centralized VCS.

1)→ It is easy to understand and get started.

2)→ You have more control over users and access (Since it is served from one place)

Date: Page:

eg, SVN

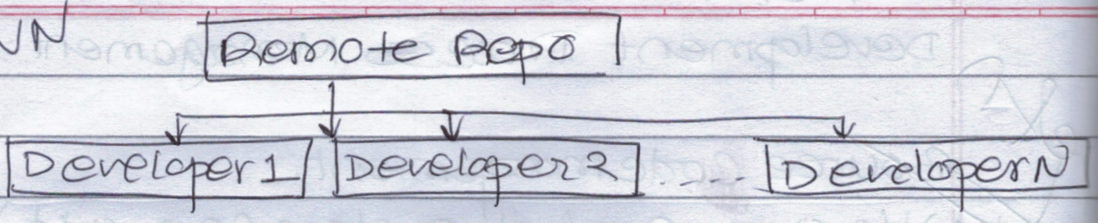


fig: centralized VCS

eg GIT

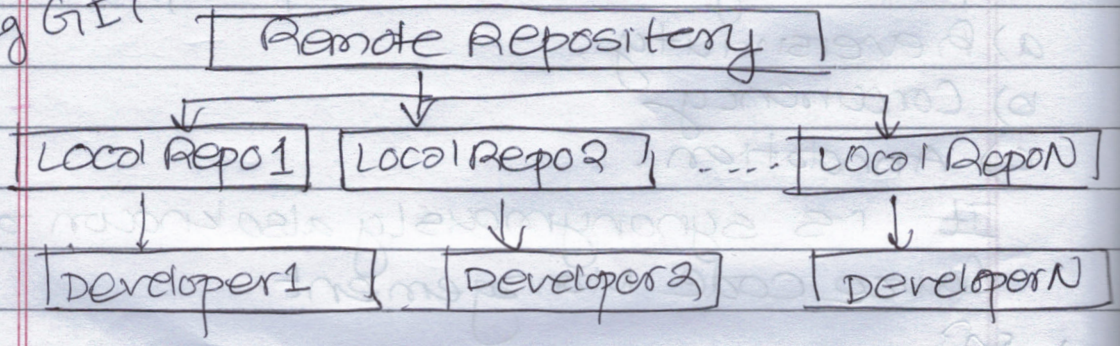


fig: Distributed VCS.

Benefits of centralized VCS.

1) → It is easy to understand and get started.

2) → You have more control over users and access (Since it is served from one place)

Date: Page:

Drawbacks of centralized VCS

a) - Dependent on access to the server
b) - hard to manage a server & backups
c) - It can be slower because every command connects to the server.
d) - Branching and merging tools are difficult to use

Benefits of Distributed (Decentralized) VCS

1) → each user has their own copy of the entire repository, not just the files but the history as well as so it is fast.
2) → More powerful and detailed change tracking which means less conflict.
3) → NO server necessary, all actions except sharing repository, are local (offline commit)
4) → Branching and merging is more reliable.

Dowbacks of Distributed VCS.

1 - The distributed model is harder to understand.

Date: Page:

Continuous Integration.

CI is a development practise that requires developer to integrate code into a shared repository, several times a day.

- Each check-in is then verified by an automated build, allowing team to detect problems early.
- By integrating regularly you can detect error quickly and locate them more easily.
- Continuous Integration is cheap, not integrating continuously is expensive
- If you don't follow a continuous approach you will have longer periods between integration.
- This makes it exponentially more difficult to find and fix problems such integration problems can easily knock off a project off schedule or cause it to fail altogether.

Date: Page:

2) Advantage of CI

a) → Increase visibility, Enabling better communication.

b) → ~~Catch~~ Catch issue early, spend less time debugging and more time adding feature.

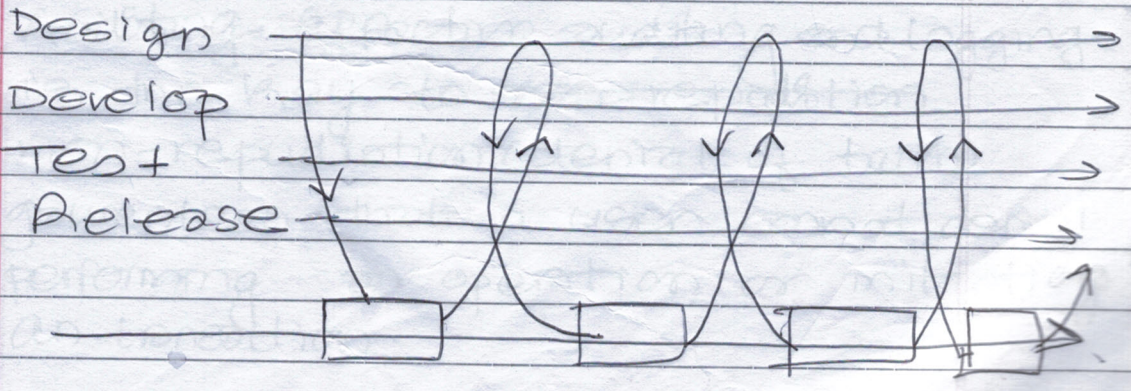
c) → Build a solid foundation.

→ Stop waiting to find out if your code is going to work.

d) → Reduce integration problems, allowing you to deliver software more rapidly

3) Making continuous integration works we need

a) Source code Repository
b) a check-in process
c) an automated build process
d) a willingness to work incrementally



GURUKUL Fig: Showing a typical CI process

Chapter-6

Date: Page:

Software security (How do you build your secure Applicati

→ SS is an idea implemented to protect software against malicious attack and other hacker risk so that the software continues to function, correctly under such potential risk.

Security is necessary to provide integrity authentication and availability

Security relies on the following elements.

1) Authentication: authentication addresses the question: who are you? It is the process of uniquely identify the clients of your applications and services.

2) Authorization: Authorization address the question: what can you do? It is the process that governs the resources and operations that the authenticated client is permitted to access.

3) Auditing: effective auditing and logging is the key to ~~non-repudiation~~ non-repudiation (denial of truth) gourentees that a user cannot denial performing an operation or initiating an transaction.

Date: Page:

4) Confidenciality: Confidenciality also refer to a privacy is a process of making sure that data remains privet and confidencial and that if cannot be viewed by unauthorized users. Encryption is frequently used to enforce confidenaality.

5) Integrity: Integrity is the gaurantee that data is protected from accidental or malicious modification. Integrity for data in transmission is typically provided by using hashing techniques and message authentication ports.

6) Availability: From the security perspective Availability system remains available for legitimate users. The goal for many attackers can denial of service attacks is to crash an application or to make that it is sufficiently overwhelmed so that other users cant access the application.

GURUKUL

Date: Page:

How do you build a secure web application

Basic Attack

1) Site Mapping : The first technique an attacker will typically use against your site is site mapping. Attacker will often do some or all of the following

→ Use a crawler to follow all the links in the application and maps outs the application page flow and authentication mechanism.

→ Read comments in the documents delivered to browser, this can give tips about business logic and algorithm. looks for usernames, passwords and database names in the documents delivers to the client.

→ looks for sql query.

→ locate hidden inputs in the forms.

→ Determine what parameters are passed between the pages.

→ Force various error scenarios and see how the server response.

#preventive measure of each attack.

Date: Page:

2) Guessing File and Directories

- file such as con.inc web.XML, server.xml can contain db connection information.....

can contain database connection info. File such as .htpasswds can contain username & password information.

3) exploiting known security flaws in support software

3

3) The next simple attack is know security flaws in web server software, operating system software and default system ll, when bugs are found, potenes are usualy posted to fix them, it takes time for system administator to apply security patches, Attackers can exploit the, time both publication of the flaw and administrator appication of the patch

GURUKUL

Date: Page:

4) Bypassing restriction on input choices.

- Another class of attacks involves, bypassing Restriction on input choices
User input widgets, normally restrict the input to well behave ranges.
- To improve the UX, client side code often performs validation of user input data.
- Poorly written application sometimes trust that, the submitted values will be drawn from the expected range.
- By bypassing the user interface, attackers can submit arbitory values, for user input variables.

Date: Page:

Cross site scripting (XSS)

- XXS attacks are a type of injection, in which malicious scripts are injected into the websites. XSS attacks occur when an attacker uses a web application to send a malicious code, generally in the form of a browser side script to a different end-user.

Types

1) Reflected / Non-persistence XSS
- occur when the data provided by a web client, most commonly in http (query parameters), is used immediately by server side scripts to pass on/ display a page of results without properly sanitizing the request.

2) Persistence
- It occurs when the data provided by the attacker is saved by the server

3)

Date: Page:

Denial of service (DOS)

A DOS attack is an attempt to shut down a machine or network, making it inaccessible to its intended users.

DOS attacks typically flood servers, systems or network with traffic in order to overwhelm the victim resources and make it difficult or impossible for ~~to~~ legitimate users ~~to~~ access them.

There are 2 general method of DOS attacks.

1) Flooding services.
2) Crashing services.

1) Flooding services.

Flood attacks occur when the system receive too much traffic for the server to buffer, causing them to slow ~~them~~ down and eventually stop.

Popular Food attacks are:

→ Buffer overflow attacks.
→ ICNP attacks. flood
→ SYN flood [synchronization flood]

Date: Page:

Signs of DOS attacks

1) Degradation in network performance especially when attempting to files stores on the network, or when accessing websites.

2) Higher then usual volume of spam e-mail

3) An inability to reach a particular websites.

#DDOS - Distributed denial of service

- An additional types of DOS is DDOS.

A DDOS attack occurs when multiple systems / plan or co-ordinate & synchronize DOS attack to a single target.

A essential difference is that instead of being attack from one location, the target is attack from many locations at once.

GURUKUL

Date: Page:

How to prevent DDOS attack?

1) By more bandwidth.
2) Build Redundancy into your ~~Infos~~ Infrastructure.
3) Configure your network hardware against ~~DDOS~~ attack.
4) Protect your DNS server
5) Deploy ~~NIT~~ Anti-DDOS software modules.

Date: Page:

State Based Attacks

Web is stateless. se.

Server doesn't care what page or what order is requested.

Web has no built in mechanism that specifies which sequence of webpages and forms ~~and~~ are presented to user.

This aspect of web is statelessness

What if our service require certain order?

What if server must keep track of things?

How can developer do this?

1) Hidden fields
2) Get or post parameters
3) Cookies
4) Session

Examples of state Based Attacks.

1) Hidden field attacks: changing the value of hidden field attacks.
2) Get or post (CGI attacks)
   common Gateway Interface

Date: Page:

Cookies poisoning
= Changing the value of the cookie

Session Hijacking.
= SH is the exploitation of a valid computer session, sometimes also called a session key to gain a unauthorizated access to information or services in a computer system. A popular method is using source routed ip packets, This allows an attacker at point B on the network to participate in a conversation between A and C by encouraging the ip packets to pass through B's machine.

GURUKUL

Date: Page:

SQL Injection-

A sql injection is an attack that consist of insertion or injection of a sql query via the input data from the client to the application.

SQL injection is the code injection technique used to data driven. in which malicious SQL stmt are inserted into an entry field for execution code injection technique

A succeful sql injection expert can read sensitive data from database, modify database data (insert / update/delete), execute administration operation (such as shut down the dbms)
SQL injection is simply looking for the faulty SQL programming.

Date: Page:

Types of SQL injection.

1) Classic SQL (In band injection)

(i) Error based: SQL is a injection technique that relies on error messages thrown by the database server to obtain info about the structure of the database.

(ii) Union based: Union based SQL injection technique forces Union SQL operator to combine the results of two or more select statements into a single results which is then returned as a response.

2) Blind SQL (Inferential SQL injection)

i) Boolean based: This type of injection technique forces the application to return a different result depending on whether the query returns true or false result.

Date: Page:

Prevention technique:

1) Prepared statement (coloin parameterized query)

→ Parameterized query force the developer to first refine all the sql code and pass ~~increase~~ each parameter to query later.
This coding style allows the database to distinguish between code and data, regardless of what user input is supplied.

→ Prepared statements ensure that an attacker is not able to change the intent of the query, even if SQL commands are inserted by an attackers.

Explanation: can't be able to able the parameters later.

2) Stored Procedure.

→ Stored procedures are not always safe from SQL injection, however certain standard, store procedure programming, constructs have the same effect as the use of parameterized queries when

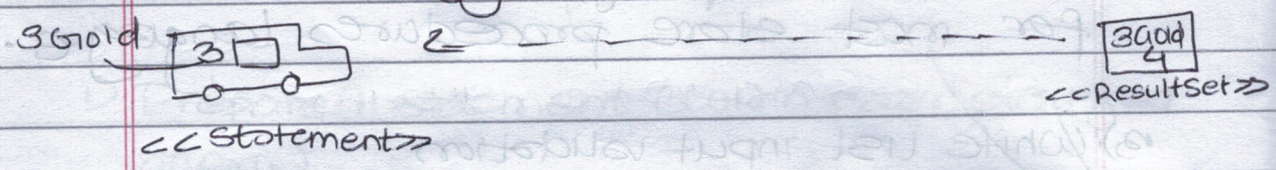
incremented safely which is the norm for most store procedures language.

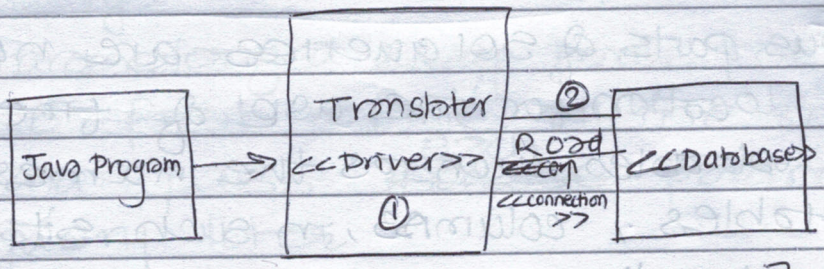
3) White list input validation.
- Various parts of SQL queries are not legal location for the use of ~~the var~~ bind variables such as the names of the tables, columns, in such situation input validation or query redesign is the most appropriate defence.

4) escaping all user supply input.

Date: _____ Page: _____

JDBC Analogy





Jwellery shop (Pokhara)

Khasa Jwellery (China)

Driver : Translator

Connection : Road

Statement : Vehicle to send sql and get Result

Resultset : Box

Date: Page:

Steps to develop JDBC Application

Step 1) : Load and Register Driver

- Class.forName("com.mysql.cj.jdbc.Driver");
  # Class.forName register the [Driver class]

Step 2): Establish connection between Java application and database

- Connection con = DriverManager.getConnection (url, username, password);

Step 3) : creation of statement object

- Statement st = con.createStatement();

Step 4): Send and execute SQL Query.

- Result rs = st.executeQuery ("select * from tablename");

Step 5) Process Result from Result set

```
while(rs.next())
{
    System.out.println(rs.getInt());
```

Step 6) : close connection

con.close();

Date: Page:

Types of SQL commands

1) DDL
- create table, alter, drop

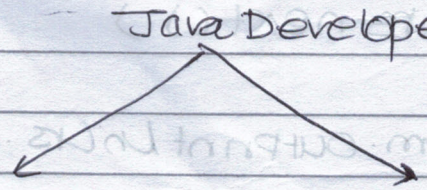
2) DML
- insert, delete, update

3) DQL
- Select

4) DCL
alter password, grant access

3) Data administrator
- start audit

6) Transaction
- commit, rollback, savepoint



DQL
→ ~~Rest~~
→ returns Resultset

DML / DDL
→ ~~Boolean~~
Numeric value

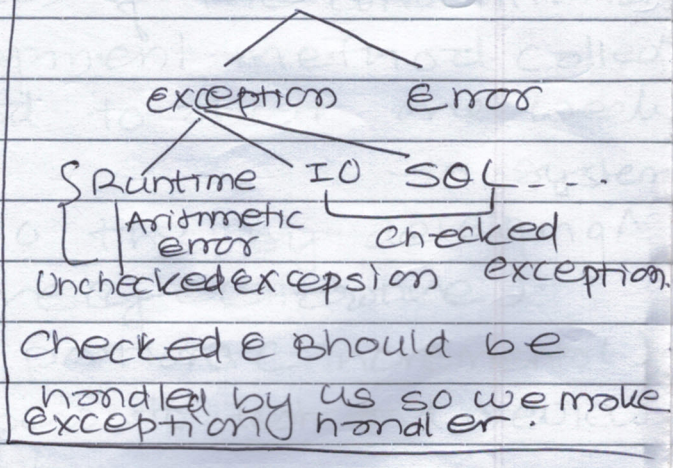
Date: Page:

Methods to execute SQL query

1) executeQuery()
- To execute select queries

→ Public ResultSet executeQuery(String sqlQuery) throws SQLException.

Not For Exam



Checked exception should be handled by us so we make exception handler.

2) executeUpdate()
→ applicable for non-select operation (DML)
→ Public int executeUpdate(String SQLQuery) throws SQL exception

3) execute();
- for both select and non-select operation.
- if you dont know SQL query until runtime.

→ public boolean execute(String query) throws SQLException
ie true → select Query
false → non select Query.

GURUKUL

Date: Page:

```
for eg
boolean b = st.execute("....");
if (b==true)
{
Requestset rs = st.getResultset();
}
else
{
int rowcount = st.getUpdatecount();
}
```

GURUKUL

Software Development Methodologies.

Chapter 7

Date: Page:

Agile methodology.

Agile software development is an approach to software development under which requirements & solution evolve through the collaborative effort of self organizing and cross functionalities and their customers/end users.

The key in agile technique is compressing the five sequences of the conventional software development method called waterfall method to about one week cycle.

It manages to do this by developing a system through repetitive cycle (iterative) and in small portions (incremental), Allowing developers to test and review during development.

Date: Page:

Agile software development principles

The ~~many~~ manifesto for agail agile SD cycle is based in 12 principles

1) Customer satisfaction by early and continues delivery of valuable software.
2) Welcome changing requirement even in late development.
3) Deliver working software frequently (weeks rather than months).
4) Close, daily co-operation beth. people and developer.
5) Projects are build around motivated individuals who should be trusted.
6) Face to face conversation is best form of communication. ~~location~~ co-location.
7) Working software is the primary measure of ~~sucess~~ progress.
8) Sustainable development ^ able to maintain a constant ~~page~~ pace. wanted
9) Continuous attention to technical excellence & good design.
10) Simplicity: The art of maximizing amount of ~~undone work~~ work not done is essential.

GURUKUL

Date: Page:

11) Best Architectures, Requirements & desings emerge from self organizing teams.

12) Regularly; the team reflects on how to became more effective & adjust accordingly

short Note.

Agile Methodologies;

1) Scrum
2) XP
3) FDD [Feature Driven Development]
4) Lean and Kanban Method]
5) DSDM [Dynamic System Development Method]
6) ASD [Adaptive software development]
7) ~~Agaile~~ Agile Unified Process.

Agile software development practices;

1) Test driven development
2) Peer programming
3) Behaviour driven development
4) Acceptance test driven development
5) Continuous Integration.

#

Date: Page:

Q1 Test Driven Development

Def a) It is a software developent process that relies on the repetition of a very short development cycle. Requirements are turned into very specific test cases - - -

b) then the software is improved to pass the new test only.

Q2 TDD is related to the test first programming concept of XP.

Q3 process

TDD ~~skills~~ starts with development of test for each one of the feature. The test might fail as the test are developed even before the development. Development team then develops and refractors the code to pass the test

~~The~~

GURUKUL

(3) SKP

Date: Page:

The following sequence is followed in TDD.

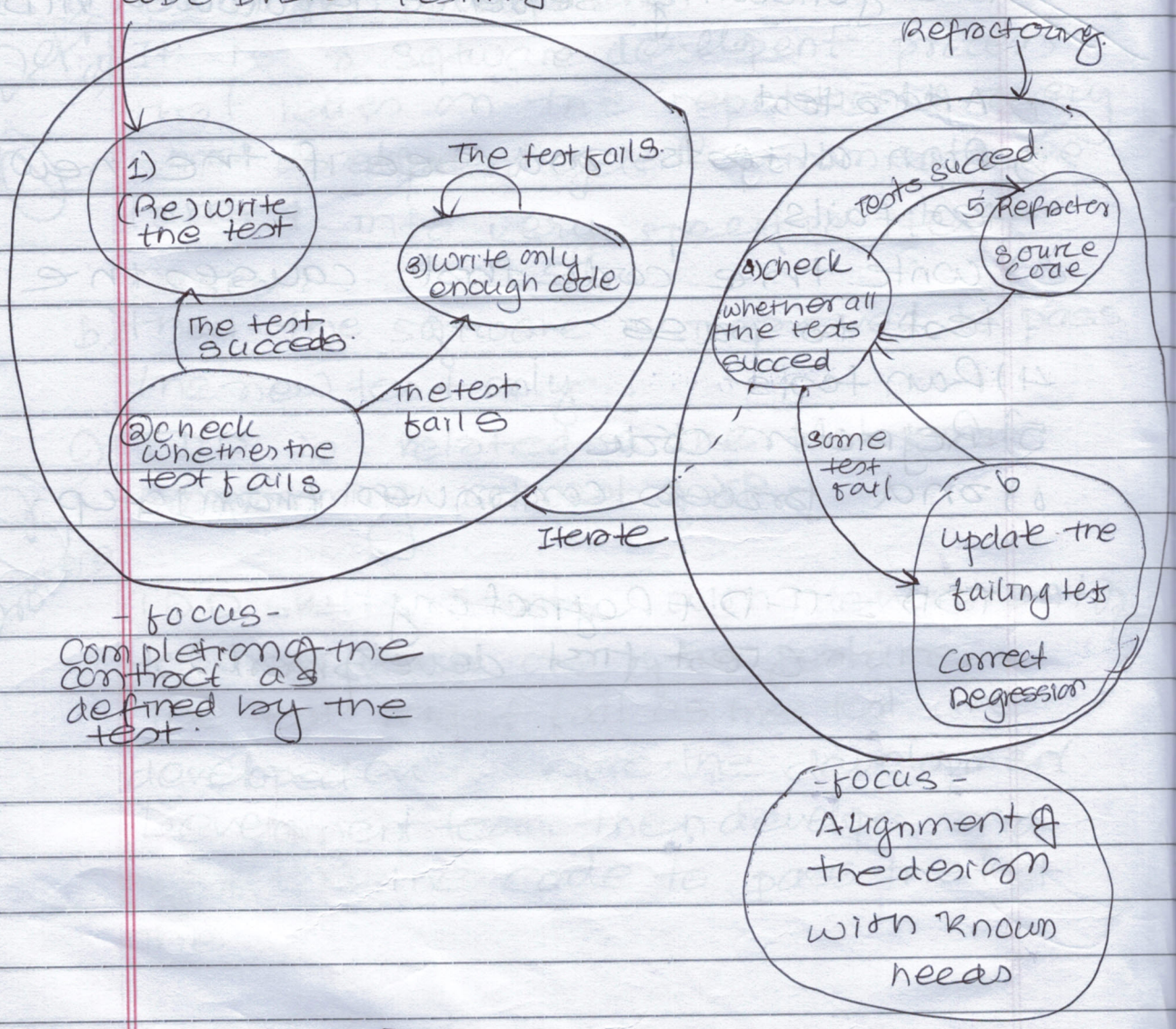
1) Add a test
2) Run all tests and see if the new test fails
3) write the code that causes the test to pass.
4) Run tests.
5) Refactor code
6) and process continues from 1 step.

TDD = TFD + Refractory
(Test first development).

GURUKUL

Date: Page:

CODE DRIVEN TESTING



- focus -
Completeness of the contract as defined by the test.

- focus -
Alignment of the design with known needs

TDD

Date: Page:

Benefits of TDD.

a) Much less debug time
b) Code proven to meets requirements
c) Test becomes safety net.
d) Near. 0 defects.
e) Shorter development cycles.

~~Peer Programming~~

Date: Page:

Behaviour Driven Development (BDD)

① BDD is sw dev. process that emerges from test Driven Dev. BDD combine the genera technique principles of TDD with ideas from Domain Driven Design and O° analysis and design to provide sw dev and mgmt, teams with shared tools and a shared process to collaborate on sw dev.

2) BDD is largely facilitated through the use of a simple domain specific language using natural language constructs that can express the behaviours & the expected outcomes.
BDD is considered an effective technical practice especially when the problem space of the business problem to solve is complex.

GURUKUL

Date: Page:

Principles of BDD

(i) BDD specifies that [tests of any unit of SW should be specified in terms of the desired behaviour of the unit].

(ii) BDD is also related to [how the desired behaviour should be specified].

(iii) BDD specified business analysis & [developer should collaborate on & should specify behaviour in terms of user stories].

Story : Returns go to stock

As a : store owner

In order to keep track of stock

I want to add items back to stock when they're returned.

Scenario 1: Refunded item should be returned to stock.

Given that a customer previously brought a black sweaters from me.

And I have there black sweaters in stock when they return the black sweater for a refund.

Then I should have 4 black sweater in a stock.

Date: Page:

Given

JBehave example // Precondition which defines the start of a scenario.

Given a 5 by 5 game
when I toggle the cell at (3,2)
// when as an event trigger
Then the grid should look like
// then as a postcondition which must be verified as the outcome of the action that follows the trigger.

when I toggle the cell at (3,1)
Then the grid should look like

- - -
- - -
- - -
- - -
- - -

- - x - -
- - x - -
- - - - -
- - - - -
- - - - -

when I toggle the cell at (3,2)
Then the grid should look like

- - - - -
- - - - -
- - - - -
- - x - -
- - - - -

GURUKUL

Date: Page:

```
class GameOfLife
{
private Game game;
private StringRender render;
@Given("a $width by $height game")
public void theGameIsRunning
        (int width, int height)
{
game = new Game(width, height);
render = new StringRender();
game.setObserver(renderer);
@when("I toggle the cell at ($column,
                    $row)");
public void iToggleTheCellAt(int column,
                    int row);
{
game.toggleCellAt(column, row);
}
@Then("the grid should look like
        $grid")
public void theGridShouldLookLike
        (String grid)
{ assest that(gameRender.asString(),
  equalTo(grid));
} }
```

GURUKUL

Chapter-4

Web App Arc

Date: Page:

Business layer: servlet
Data layer: JPA, JDBC
Presentation layer: HTML, CSS, JS, JSP

Q How do you design Business logic in your app using java technology.

1) # What is servlet?

Q) - It is a technology that is used to create enterprise web application.

2) - servlet is an API that provides many interface & classes in loading documentation.

3) - servlet is an interface that must be implemented for creating any ~~servlet~~ ~~service~~ serlet
- It extends the capabilities of the server & responds to the incoming requests.
- It is a web component that is deployed on the server to create a dynamic web application.

GURUKUL

(2)

Date: Jax.S.*) J.S.http.X Page:

servlet API

1) Javax.servlet.*;
2) Javax.servlet.http.*;

(3)

some pacbages and interface

i) servlet(i) → javax.servlet.*.
ii) " Request(i) → " " "
iii) " Dispatchers(i) → " " "
iv) Generic servlet(abstract_class) → javax.serlet.x,
v) Httpservlet[abstract(i) → javax.servlet.http.*
vi) Httpservelet Request(i) → javax.servelet.http.*
vii) " " Response(i)) → " "
ix) FiHa(i) → javax.servlet.*

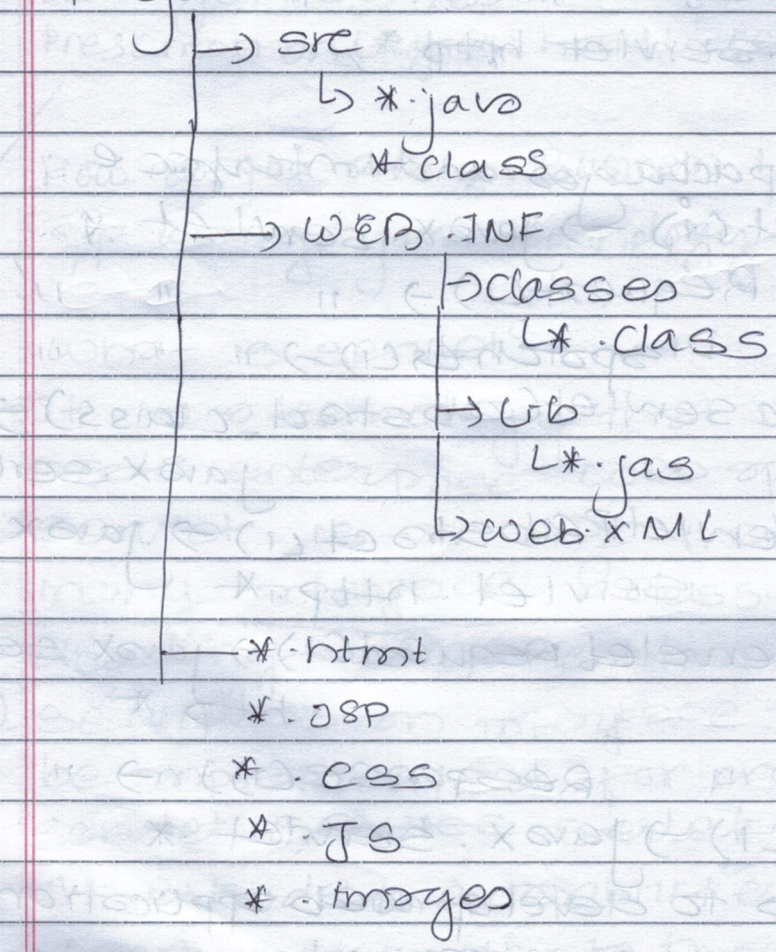
steps to develop web appucation.

Imp
Step 1: Develop the webappucation architecture
" 2: Develop " web resources (HTML file, CSS file, servlet file etc).
Step3: Develop the deployment descriptor file(web.XML)
Step4: Compile the java file
" 5: Deploy appucation into server
" 6: start server and access using web browser.

GURUKUL

Date: Page:

Q) # web app. war file folder structure.



Servlet(I) life cycle method.

1) public void init(ServletConfig config) throws ServletException

a) It is called only once.

b) It is called only when the servlet is creator & not called for any user request & afterward.

Date: Page:

- The init() method simply creates or loads some data that will be used throughout the life of the servlet

2) Public void service(ServletRequest req, ServletResponse res) throws ServletException, IOException.

a) - It is the main method to perform the actual task.

b) - The servlet container (i.e. webserver) calls to service() to handle request coming from the client and write the formatted response back to the client.

c) Each time the server receives request for a servlet the server create new thread and call service

d) It checks the HTTP Request type (GET, POST, PUT) and calls doGet(), doPost(), doPut(), doDelete()

e) So you have nothing to do with service(), you to just override doGet(), doPost().. etc.

Date: Page:

3) Public void destroy()

It is called only once at the end of the lifecycle of a servelet.

This method gives your servlet a chance to close db connection halt background, threads, perform cleanup etc.

(Chapter-5) Date: Page:

Soap VS RESTful webservices.

SOAP:

1) A XML-based message protocol
2) Uses WSDL for communication between consumer and provider
3) Invokes services by calling RPC method.
4) Doesn't Return human Readable Result
5) Transfer is over HTTP, also uses other protocols such as SMTP, FTP etc
6) Javascript can't call SOAP, but it is diffical to implement.
7) Peyormance is not great compared to REST

~~SOAP~~
Soap
→ ~~Apache CXF-REST~~
→ ~~Jersey 2.x~~
→ ~~Spring~~

SOAP
Apache CXF-SOAP

GURUKUL

8) Only contract is exposed to outside world through WSDL document which could be any operation (depends on underlying developer or designer implementation.

9) Generally used in large enterprise applications like Telecom, Finance etc.

10) Various vendor implementation for SOAP webservices
→ Apache ~~CFS~~ AXIX - 2RI
- Apache CXF - SOAP

GURUKUL

Restful

Date: Page:

1) An architectural architectural style protocol
2) Uses XML or JSON to send and Recieve data
3) Simply calls servics via URL path.
4) Result is readable which is just plain XML or JSON
5) Transfer is over HTTP only
6) Easy to call from Javascript
7) Performance is much better compared to SOAP.

8) For different CRUD operation like, insert/select/update/delete, different HTTP methods are available.
POST, GET, PUT, DELETE

9) Used for light-weigh applications like mobile application, ~~which~~ where processing required at very high speed

10) Various vender implementations for Rest webservices are.
- Apache CXF - REST
- Jersey 2.x
- Spring Restful.

Date: Page:

Payload in Restful

- A payload is a data sent over the internet and when a payload is 'heavy' it requires more resources. Rest tends to use HTTP & JSON which lighten the payload.

Restful component architecture

- lets consider, I need to develop a REST Bank application that allows [creation/deletion] of bank accounts as well as withdraw/credit/getBalance (pure)

i) Creation of account.

PUT/Bank/john

ii) Destruction of Account
DELETE/Bank/john

iii) Get Balance
GET/Bank/john

iv) withdraw money
POST/Bank/john
action=withdraw & value=10

V. Imp for Exam.

Date: Page:

MVC example & [Business, Presentation, Data layer].

To implement webapplication based on MVC design pattern

i) Create Student & studentservice class : Modellayer [Data source layer];

ii) Student Servletclass : Controller [Business

iii) Student_record.jsp : Presentation. view]

Student class

```
public class Student
{ private int id;
  private string firstName;
  private string lastName;
}
```

Student service class

```
public class StudentService{
public Optional<Student>
   getstudent(int id)
```

Date: Page:

Student Service class

```
public class StudentService {
  public Optional<Student> getStudent(int id)
  {
   Switch(id) {
   case 1 :
      return Optional.of(new Student
            (1, "John", "Doe" )
   case 2 :
   return Optional.of(new Student (
         2, "Jane", "Will" ))
   default :
      return Optional.empty();
   }
  }
}
```

Date: Page:

11 Studentservlet class

```
@webservlet (
name = "StudentServlet",
urlPatterns = "/student-record" )

Pudic class Studentservelet extends
    Httpservlet {

private studentservice studentservice
    = new studentservice ();

pirvate void proces Request (
HttpservletRequest orequest,
HttpserveletResponse oresponse)

throws servletexception, IOexception {



}
protected void doGet() {
}
protected void doPost() {
}
```

Date: Page:

Student-record.jsp

```
<html>
<head>
    <title>Student Record</title>
</head>
<body>
    <% Student student = (Student) request.getAttribute("student");
    %>
    <h1> Student Record </h1>
    <div>ID : <%= student.getId() %>
    </div>
    <div> First Name : <%= student.getFirstName() %> </div>
    <div> Last Name : <%= student.getLastName() %> </div>
</body>
</html>
```

Date: Page:

Design considerations of developing good web application.

① For Presentation layer

i) Use the Relevant pattern
ii) Design for separation of concerns.
iii) Consider human interface guidelines
iv) Adhere to use-driven design principles.

2) Business Layer

i) Applicate Facade
ii) Business components
iii) Business entity components
iv) Business workflow.

3) Data Access layers

i) Choose the data access technology
ii) Consider security risks.
iii) Use abstraction to implement a loosely coupled Interface to the data access layer.
iv) Decide how you will handle data exception.
v) Decide how you will manage connection.

V. IMP CP - 1

Date: Page:

HTTP is a stateless protocol:

1) A stateless protocol doesn't require server to Retain Information or status about each user for the duration of multiple request.

2) HTTP has HTTP cookies. Cookies allow the server to track the user state, the no of connections, last connection et.

3) HTTP has a persistent connection (keep-alive) where several request can be sent from the same TCP connection.

4) But some web Apps may have to track the user's progress from page to page.
Solution of for these cases
   a) use of HTTP cookies
   b) Server-side sessions
   c) Hidden variables
   d) URL-Rewriting using URL-encoded parameters.