# Software Engineering

Assignment 2

**Q1:**

The approaches that support software reuse are:

* Application frameworks
  + Application frameworks gives a standard way to build and deploy applications. It's an abstraction in which software providing generic functionality can be selectively changed by additional user-written code, thus providing application-specific software and achieving the software reuse concept.
  + Examples/Applications:
    - .Net Application frameworks
    - Software development kits (SDKs) for web and android app development are also an example of Application frameworks.
* Software product lines
  + It refers to the tools and techniques for creating a collection of similar software systems from a shared set of software assets using a common means of production.
  + Examples/Applications:
    - Samsung's product line of software product chain for making variety of Mobile phones.
    - GNU compiler suite.
* Application system/COTs integration
  + In it, two or more application systems are integrated to provide extended functionality.
  + Examples/Applications:
    - This approach is used when there is no COTs system that fulfils the complete requirements so 2 or more COTs are combined. (COTs are software or hardware products that are ready-made and available for sale to the general public i.e Microsoft office package)
    - E-procurement system.
* ERP Systems
  + ERPs are large scale systems that encapsulate generic business functionality and rules which are configured for an organization.
  + Examples/Applications:
    - It's commonly used in Supply Chain industries and configured accordingly.There is also an opportunity to reuse experience from one ERP implementation project to another.
    - Names of some commonly used ERPs are: Oracle ERP Cloud, Oracle NetSuite ERP, Microsoft dynamics.
* Configurable application Systems
  + In it, domain specific systems are designed so that they can be configured according to the needs of specific system customers.
  + Examples/Applications:
    - Laboratory Information Management Systems (LIMS)
    - Data Acquisition Systems.
    - Supervisory Control And Data Acquisition (SCADA)
* Design patterns
  + Patterns support high-level, concept reuse.
  + Using patterns means that you reuse the ideas but can adapt the implementation to suit the system that you are developing.
  + Examples/Applications:
    - Abstract form of domain model or class diagram can be reused by making some extension using the concept of inheritance.
* Service oriented systems:
  + Systems are developed by linking shared services that may be externally provided.
* Component based reuse:
  + Systems are developed by integrating components (collections of objects) that conform to component-model standards.
  + Examples/Applications:
    - In several web/android apps, UI functional components are frequently reused with small variations in their design.

**Q2:**

**Legacy System Reuse**

Legacy system code is wrapped behind an XML shell which allows individual functions within the programs, to be offered as web services to any external user. By means of this wrapping technology, a significant part of the company software assets can be preserved within the framework of a service oriented architecture. Therefore, by 'wrapping' the legacy system, a set of interfaces is defined, providing access to legacy system through these interfaces.

**Q3:**

**Web service module reuse:**

One of the example for achieving the web service module reuse is 'reusable rest APIs' which connects the backend data sources to the multiple client application having similar needs. So, there’s no need to keep building server-side software for each new application project.

The open source movement has meant that there is a huge reusable code base available at low cost.This may be in the form of program libraries or entire applications. There are many domain-specific application systems available that can be tailored and adapted to the needs of a specific company. Some large companies provide a range of reusable components for their customers. Standards, such as web service standards, have made it easier to develop general services and reuse them across a range of applications.

Another good example of modular reuse in web services is XML Schema.

Scenario:

When teams are developing number of apps and tie different APIs on different data bases then they start to confuse the data access rights and therefore management becomes a giant mess at this time, reusable rest web APIs can be the best strategy that supports general-purpose application development by having consistent URL structure for accessing any backend data source where, parameter names should be reused across services where possible. This presents developers with a familiar interface for any data source.

**Q4:**

Software components that implement a single function,such as a mathematical function, or an object class may be reused. You reuse the classes and functions in these libraries by linking them with newly developed application code. In areas such as mathematical algorithms and graphics, where specialized expertise is needed to develop efficient objects and functions, this is a particularly effective approach.

.