

**LEARNMORE NDLOVU**

**INFORMATION TECHNOLOGY**

**H160170B**

**PLATFORM TECHNOLOGIES  
ASSIGNMENT 2**

1. **Explain Key components of the Android Application framework**

Activity Manager − Controls all aspects of the application lifecycle and activity stack.

Content Providers − Allows applications to publish and share data with other applications.

Resource Manager − Provides access to non-code embedded resources such as strings, color settings and user interface layouts.

Notifications Manager − Allows applications to display alerts and notifications to the user.

View System − An extensible set of views used to create application user interfaces.

1. **Explain the following terms giving examples and when they can be used**

Pull

If you make a change in a repository, GIT PULL can allow others to view the changes. It is used to acknowledge the change that you've made to the repository that you're working on. Or also called a target repository.

The simple command to PULL from a branch is:

git pull 'remote\_name' 'branch\_name'.

Fork

Fork is a copy of a repository that allows you to freely experiment with changes without affecting the original project. A forked repository differs from a clone in that a connection exists between your fork and the original repository itself.

Push

The git push command is used to transfer or push the commit, which is made on a local branch in your computer to a remote repository like GitHub. The command used for pushing to GitHub is given below.

git push 'remote\_name' 'branch\_name'

1. **State four iOS weaknesses**

* Not flexible only supports iOS devices
* Applications are very large when compared to other mobile platforms
* Using iOS are costly Apps and no widget support
* Repair costs are very piracy

1. **What do you understand by breaking a smartphone?**

It is a process which bypasses Digital Rights Management (DRM) technologies on a cell phone lifting any media restrictions. It grants super user permissions that is giving access to advanced options that would be unavailable and at the same time exposing it to malicious attack and also allows phone holders to download and use software and apps they want that are not available in the Apple App Store or Google Play Store.

1. **List and explain 3 examples of version control systems.**

Github

Git is the new fast-rising star of version control systems. Initially developed by Linux kernel creator Linus Torvalds, Git has recently taken the Web development community by storm. Git offers a much different type of version control in that it’s a distributed version control system. With a distributed version control system, there isn’t one centralized code base to pull the code from. Different branches hold different parts of the code. Other version control systems, such as SVN and CVS, use centralized version control, meaning that only one master copy of the software is used.

SVN

Subversion is probably the version control system with the widest adoption. Most open-source projects use Subversion as a repository because other larger projects, such as SourceForge, Apache, Python, Ruby and many others, use it as well. Google Code uses Subversion exclusively to distribute code.

Because of Subversion’s popularity, many different Subversion clients are available. If you’re a Windows user, Tortoise SVN is a great file browser for viewing, editing and modifying your Subversion code base. If you’re on a Mac, Versions is an elegant client that provides a “pleasant way to work with Subversion.” Xcode is Apple’s developer environment and Subversion client that ships with Leopard on a Mac.

Mercurial

Mercurial is another open-source distributed version control system, like Git. Mercurial was designed for larger projects, most likely outside the scope of designers and independent Web developers. That doesn’t mean that small development teams can’t or shouldn’t use it. Mercurial is extremely fast, and the creators built the software with performance as the most important feature. The name “mercurial” is an adjective that means “Relating to or having characteristics (eloquence, swiftness, cleverness) attributed to the god Mercury.”

1. **Describe activities that are taken during software deployment.**

Release

A software system’s release is the initial activity in the systems deployment process. It takes place on producer site after the system’s development has been completed. The system is packaged so that it can be transferred to the consumer sites.

Install

Once the system has been packaged at the producer site and transferred to the consumer site, it is ready to be configured and installed for operation.

Activate

Once the system is installed, it needs to be activated for use on the target hosts. Activation consist of providing a command, or sequence of commands that will be required to start up the system.

Deactivate

It involves disabling or shutting down a system of any of the system’s facilities that are still active on the target hosts.

Update

Once the system has been installed and activated on target hosts, it may need to be updated overtime for different reasons. The system may need to be deactivated before it is updated and thereafter re-activated. Updates are initiated by the system’s producers and involve the same activities as the original installation. It is critical that the update must be properly reflected in its architectural models. If it is not ensured, the architecture will degrade and any subsequent updates may result in system defect.

Adapt

Adaptation encompasses a wide range of activities and results in dynamically changing the system in response to events in the system’s execution environment.

De-install

If the system is no longer needed on the consumer sites, it is removed. The deinstallation is done by simply reversing the steps taken during installation. It should be noted that, before the system is de-installed, it may need to be deactivated first.

De-release

The producer of the given system may decide not to support the system or retire the system any longer i.e. the producer may decide to retire the system. This may be because of evolution of a superior version of that software or market size is too small or the producer has discontinued the product and gone out of business etc.

1. **State and explain 2 applications of embedded systems. Give an example of each.**

Stand Alone Embedded Systems

Stand alone embedded systems do not require a host system like a computer, it works by itself. It takes the input from the input ports either analog or digital and processes, calculates and converts the data and gives the resulting data through the connected device-Which either controls, drives and displays the connected devices. Examples for the stand alone embedded systems are mp3 players, digital cameras, video game consoles, microwave ovens and temperature measurement systems.

Real-time embedded system:

Real-time systems are those which give a quick response to critical situations. They are used in military, medical and industrial applications. Engineers working in these systems have high demand is current days. To develop the real-time embedded system we require timing analysis, multitasking design, debugging, cross-platform testing and architecture design. In these systems, quick response is very important. Better hardware is also used in these systems to avoid failure in performance. Real-time systems control the external environment by input & output interfaces and sensors. The external environment includes human and other animals. Some examples of real-time embedded systems include:-

Controlling heat, elevators, lights, and doors in buildings

Robots

Traffic control system including railway tracks, airspace, shipping lines, highways

Radio, satellite and telephone communication

1. **Compare and contrast RESTful and Soap services**

|  |  |
| --- | --- |
| SOAP | REST |
| SOAP is a **protocol**. | REST is an architectural style. |
| SOAP stands for Simple Object Access Protocol. | REST stands for **Representational State Transfer**. |
| SOAP **uses services interfaces to expose the business logic**. | REST **uses URI to expose business logic**. |
| JAX-WS is the java API for SOAP web services. | **JAX-RS** is the java API for RESTful web services. |
| SOAP **can't use REST** because it is a protocol. | REST **can use SOAP** web services because it is a concept and can use any protocol like HTTP, SOAP. |
| SOAP **permits XML** data format only. | REST permits different data format such as Plain text, HTML, XML, JSON etc |
| SOAP defines its own security. | RESTful web services inherits security measures from the underlying transport. |
| Requires more resources and bandwidth | Requires fewer resources and is lightweight |
| SOAP defines standards to be strictly followed. | REST does not define too much standards like SOAP. |