**a)** All software are programs, which are also called **source packages**. So all source packages need to be **built** first, to run on your system.

The **binary packages** are one that are already build from **source** by someone with general features and parameters provided in the software so that a large number of users can install and use it.

**b)** Installing from **source** means you compile the souce code and then install.

Installing from **binary** means someone allready compiled it into binary and you just have to put it on your computer.

**c)** **Source packages** are simply packages which just include source code, and can generally be used on any type of machine if the code is compiled in the right way. (For information on how to compile and install source packages, see CompilingEasyHowTo).

**Binary packages** are ones which have been made specifically for one type of computer, or architecture. Ubuntu supports the x86 (i386 or i686), AMD64 and PPC architectures. The correct binary packages will be used automatically, so you don't have to worry about picking the right ones.

d)

Basically, **Source** packages need to be compiled by you before installing (they are the code itself) (using make and a compiler).

**Binary** packages are already compiled for your achitecture and only need to be installed (using **dpkg** for example)

**ADVANTAGES & DISADVANTAGES**

**1. Binary:**  Easy to install but you cant customize the software acc to your need.

**2. Source:** Complex to install but you can customize eg add/trim features

**7.1 What is a Debian package?**

Packages generally contain all of the files necessary to implement a set of related commands or features. There are two types of Debian packages:

**Binary packages,** which contain executables, configuration files, man/info pages, copyright information, and other documentation. These packages are distributed in a Debian-specific archive format (see What is the format of a Debian binary package?, Section 7.2); they are usually distinguished by having a **'.deb'** file extension. Binary packages can be unpacked using the Debian utility dpkg (possibly via a frontend like aptitude); details are given in its manual page.

**Source packages,** which consist of a .dsc file describing the source package (including the names of the following files), a .orig.tar.gz file that contains the original unmodified source in gzip-compressed tar format and usually a .diff.gz file that contains the Debian-specific changes to the original source. The utility dpkg-source packs and unpacks Debian source archives; details are provided in its manual page. (The program apt-get can get used a frontend for dpkg-source.)

Installation of software by the package system uses **"dependencies"** which are carefully designed by the package maintainers. These dependencies are documented in the control file associated with each package. For example, the package containing the GNU C compiler (gcc) "depends" on the package binutils which includes the linker and assembler. If a user attempts to install gcc without having first installed binutils, the package management system (dpkg) will send an error message that it also needs binutils, and stop installing gcc.

**7.11 What is meant by unknown, install, remove, purge and hold in the package status?**

These **"want"** flags tell what the user wanted to do with a package (as indicated either by the user's actions in the "Select" section of dselect, or by the user's direct invocations of dpkg).

Their meanings are:

**unknown** - the user has never indicated whether he wants the package

**install -** the user wants the package installed or upgraded

**remove** - the user wants the package removed, but does not want to remove any existing configuration files.

**purge** - the user wants the package to be removed completely, including its configuration files.

**hold** - the user wants this package not to be processed, i.e., he wants to keep the current version with the current status whatever that is.

There are three ways of holding back packages, with **dpkg**, **aptitude** or with dselect.