www.vskills.com

Certified Linux Administrator VS-1064

Vskills Certifications

Vskills Reading Material

Skills for a secure future

Certified Linux Administrator

VS-1064

V skills

www.vskills.in

This document describes the Linux operating system which covers areas of OSS, installation, kernel and file system, shell scripting, networking, user management and advanced topics like virtualization. This document is for beginners and intermediaries.

Certified - Linux Administrator

Training Material

1. LINUX INTRODUCTION

An Operating System is a large collection of software, which manages resources of the computer system, such as memory, processor, file system and input/output device. It keeps track of the status of each resource and decides who will have a control over computer resources.

An Operating System is a set of programs that control the operations of a central processing unit (CPU) and enable the user to communicate with the system. The major ones for microcomputers are DOS, WINDOWS and UNIX.

Linux is an operating system, a large piece of software that manages a computer. It is similar to Microsoft Windows, but it is entirely free. The accurate name is GNU/Linux but "Linux" is used more often. An Operating System is a large collection of software, which manages resources of the computer system, such as memory, processor, file system and input/output device. It keeps track of the status of each resource and decides who will have a control over computer resources.

Linux is not one company's product, but a number of companies and groups of people contribute to it. In fact, the GNU/Linux system is a core component, which is branched off into many different products. They are called distributions.

Distributions change the appearance and function of Linux completely. They range from large, fully supported complete systems (endorsed by companies) to lightweight ones that fit on a USB memory stick or run on old computers (often developed by volunteers).

Parts of Linux OS

The Kernel - All operating systems have kernels, built around the architectural metaphor that there must be a central set of instructions to direct device hardware, surrounded by various modular layers of functionality. The Linux kernel is unique and flexible because it is also modular in nature.

Modularity is desirable because it allows developers to shed parts of the kernel they don't need to use. Typically a smaller kernel is a faster kernel, because it isn't running processes it does not need.

If a device developer wants a version of Linux to run on a cell phone, she does not need the kernel functionality that deals with disk drives, Ethernet devices, or big monitor screens. She can pull out those pieces (and others), leaving just the optimized kernel to use for the phone.

The kernel of the Window operating system (which few people outside of Microsoft are allowed to look at without paying for the privilege) is a solidly connected piece of code, unable to be easily broken up into pieces. It is difficult (if not impossible) to pare down the Windows kernel to fit on a phone.

This modularity is significant to the success of Linux. The ability to scale down (or up) to meet the needs of a specific platform is a big advantage over other operating systems constrained to just a few possible platforms.

Modularity also effects stability and security as well. If one piece of the kernel code happens to fail, the rest of the kernel will not crash. Similarly, an illicit attack on one part of the kernel (or the rest

of the operating system) might hamper that part of the code, but should not compromise the security of the whole device.

Library and Utilities - Developers need special tools (like the compilers and command lines found in GNU) to write applications that can talk to the kernel. They also need tools and applications to make it easy for outside applications to access the kernel after the application is written and installed.

This collective set of tools, combined with a kernel, is known as the operating system. It is generally the lowest layer of the computer's software that is accessible by the average user. General users get to the operating system when they access the command line. These tools are

- ✓ The system libraries contain program methods for developers to write software for the operating system. The libraries contain methods for process creation and manipulation, file handling, network programming, etc.
  It is a vital part of an operating system because you can't (or shouldn't) communicate with the kernel directly: the library shields off the complexity of kernel programming for the system programmer. One example standard is the C library, probably the most important system library available. This library makes pretty vital operations available to the programmer, such as basic input/output support, string handling routines, mathematical methods, memory management and file operations. With these functions a programmer can create software that builds on every operating system that supports the C library. These methods are then translated by the C library to the kernel specific system calls (if system calls are necessary). This way the programmer doesn't need to know the kernel internals and can even write software (once) that can be build for many platforms. There is no single specification on what a system library is. The author of this book believes that system libraries are whatever library is part of the default, minimal install of an operating system. As such, system libraries for one operating system (and even Linux distribution) can and will differ from the system libraries of another. Most Linux distributions have the same system libraries, which is to be expected because all Linux distributions can run the same software and this software is of course built for these system libraries. Some distributions just don't mark one library part of the default, minimal install while others do. The most well-known system library for Linux systems is the GNU C Library, also known as glibc.
- ✓ The system tools are built using the system libraries and enable administrators to administer the system: manage processes, navigate on the file system, execute other applications, configure the network.

  But what are system tools? Well, with a kernel and some programming libraries you can't manipulate your system yet. You need access to commands, input you give to the system that gets interpreted and executed. These commands do primitive stuff like file navigation (change directory, create/remove files, obtain file listings, ...), information manipulation (text searching, compression, listing differences between files, ...), process manipulation (launching new processes, getting process listings, exiting running processes, ...), privilege related tasks (changing ownership of files, changing user ids, updating file permissions, ...) and more. If you don't know how to deal with all this stuff, you don't know how to work with your operating system. Some operating systems hide these tasks behind complex tools, others have simple

tools for each task and bundle the power of all these tools. Unix (and Linux) is one of the latter. Linux systems usually have the GNU Core Utilities for most of these tasks.

- ✓ The development tools provide the means to build new software on (or for) the system. Although not a required part of an operating system I do like to mention it because with Gentoo, this is a requirement. These tools include compilers (translate code to machine code), linkers (which collect machine code and bring it together into a working binary) and tools that ease the build process considerably.

  These additional tools and libraries are of course written by programmers and they must be able to build their code so that it works on your system. Some systems, like Gentoo Linux, even build this software for you instead of relying on the prebuilt software by others. To be able to build these tools, you need the source code of each tool and the necessary tools to convert the source code to executable files. These tools are called a tool chain: a set of tools that are used as in a chain in order to produce a working application. A general tool chain consists out of a text editor (to write the code in), compiler (to convert code to machine-specific language), linker (to combine machine-code of several sources - including prebuilt "shared" libraries - into a single, executable file) and libraries (those I just mentioned as being "shared" libraries). A tool chain is of the utmost importance for a developer; it is a vital development tool, but not the only development tool. For instance, developers of graphical applications generally need tools to create graphics as well, or even multimedia-related tools to add sound effects to their program. A development tool is a general noun for a tool that a developer would need in order to create something, but isn't vital for an operating system of an average non-developer user. The most well-known development tools are also delivered by the GNU foundation, namely the GNU Compiler Collection, also known as gcc.

The Environment - The windows, menus, and dialog boxes most people think of as part of the operating system are actually separate layers, known as the windowing system and the desktop environment.

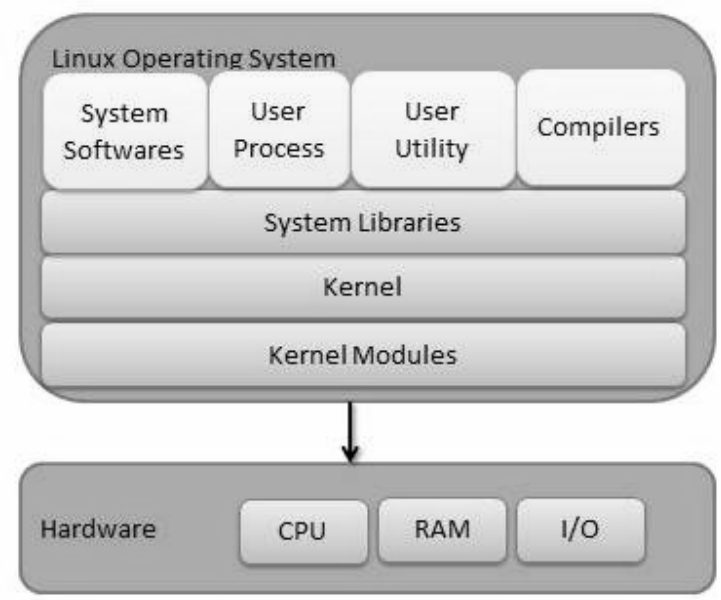
These layers provide the human-oriented graphical user interface (GUI) that enables users to easily work with applications in the operating system and third-party applications to be installed on the operating system.

In Linux, there a lot of choices for which windowing system and desktop environment can be used, something that Linux allows users to decide. This cannot be done in Windows and it's difficult to do in OS X.

Like the operating system and kernel, there are tools and code libraries available that let application developers to more readily work with these environments (e.g., gtk+ for GNOME, Qt for KDE).

The Applications - Operating systems have two kinds of applications: those that are essential components of the operating system itself, and those that users will install later. Closed operating systems, like Windows and OS X, will not let users (or developers) pick and choose the essential component applications they can use. Windows developers must use Microsoft's compiler, windowing system, and so on.

Linux application developers have a larger set of choices to develop their application. This allows more flexibility to build an application, but it does mean a developer will need to decide which Linux components to use.



Files and processes

Everything in LINUX is either a file or a process. A process is an executing program identified by a unique PID (process identifier). A file is a collection of data. They are created by users using text editors, running compilers etc. Examples of files:

- ✓ a document (report, essay etc.)
- ✓ the text of a program written in some high-level programming language
- ✓ instructions comprehensible directly to the machine and incomprehensible to a casual user, for example, a collection of binary digits (an executable or binary file);
- ✓ a directory, containing information about its contents, which may be a mixture of other directories (subdirectories) and ordinary files.

1.1. Linux and OSS

When you get a distribution of GNU/Linux, you also get the freedom to study, copy, change, and redistribute it – that's what makes it truly free software.

By virtue of its open source licensing, Linux is freely available to anyone. However, the trademark on the name "Linux" rests with its creator, Linus Torvalds. The source code for Linux is under copyright by its many individual authors, and licensed under the GPLv2 license. Because Linux has such a large number of contributors from across multiple decades of development, contacting each individual author and getting them to agree to a new license is virtually impossible, so that Linux remaining licensed under the GPLv2 in perpetuity is all but assured.

Linux is also unique from other operating systems in that it has no single owner. Torvalds still manages the development of the Linux kernel, but commercial and private developers contribute other software to make the whole Linux operating system.

Copyright

It is the law that protects the rights held by the creator, developer, or author over their own original work. These works may consist of electronically stored words, photograph, music, work of visual art, or performance art and thereby includes digital property.

There are two types of copyright infringement

- ✓ Digital Piracy: Unauthorized reproducing (copying) and distributing (sharing) of digitized property: electronic files and audio-visual media. Obtaining copies of songs, computer software, videos games, and movies without paying for them is all considered Digital Piracy.
- ✓ Plagiarism: The copyright violation of using printed words, photos, or illustrations in your own product without permission - where one person copies another's work and uses it as their own.

Copyright protection includes literary works such as novels, poems and plays, films, musical works, artistic works such as drawings, paintings, photographs and sculptures and architectural designs. Rights related to copyright include those made by artists in their performances of sound recordings and television in its radio and television broadcasts.

License

Licenses are granted by an authority to allow a usage; the use and distribution of resources by the copyright owner. I may decide to offer my photograph for free or charge a price; either way, I can include a license to limit usage, and I maintain the copyright. Just because someone pays money doesn't mean they have full control or rights to what they're buying. Licenses can dictate the number or uses, the bounds of use and even the length of time until the license expires.

When we create something like a photograph or write a book, we own the copyright, which is our exclusive right as the author to own that work. We control who else can use our work and in what manner. For example, I could allow someone to print the book or adapt it for a movie. Instead of verbal agreements, it is distributed with a license that sets the guidelines for use. The things that are copyrighted are also referred to as intellectual property.

A license can be written from scratch, but most people choose a well-known one. Some common terminology used in licensing, is

- ✓ "Copy" - A simple copy of the original work.
- ✓ "Modify" - To alter copyrighted work in some way before using it.
- ✓ "Derivative work" - The result of modifying copyrighted work to produce new work.
- ✓ "Distribute" - The act of giving someone your work under a license.
- ✓ "Redistribute" - The act of distributing work and its license after obtaining it under license from the original copyright owner.
- ✓ "Share alike" - Permission to distribute derivative work under the same or a similar license.
- ✓ "Credit" or "attribution" - The act of identifying the original copyright owner.

- ✓ "Copyright notice" - A written phrase or symbol (©) informing of copyright ownership (not necessarily required by law).
- ✓ "All rights reserved" - A common copyright notice declaring that no usage rights exist (again, not necessarily required).
- ✓ "Warranty" - A written guarantee included with the license (or, usually, not).

Creative Commons - Lawrence Lessig founded Creative Commons (CC) in 2001 to create a series of easy-to-understand copyright licenses for online creative work. These licenses established the notion of "some rights reserved."

The Creative Commons license has six variations. It's really a collection of licenses that cover particular uses. These include whether the licensed work can be used commercially, whether it can be modified, and whether derivative work can be redistributed under the same (or a compatible) license. A Creative Commons license can be restricted to certain jurisdictions or apply internationally.

The basic Creative Commons license is CC Attribution. It allows for all copying, modification and redistribution (even commercially), provided that the original author is attributed (with no implication of endorsement). Work under CC Attribution is essentially free to use.

The CC Attribution license can be extended to CC Attribution-ShareAlike. The same rules apply, except that all derivative work must be licensed the same way. This distinction ensures that all resulting work remains free. Wikipedia uses this license for its text.

Software License

It is a legal instrument (usually by way of contract law, with or without printed material) governing the use or redistribution of software. Under United States copyright law all software is copyright protected, except material in the public domain. A typical software license grants an end-user permission to use one or more copies of software in ways where such a use would otherwise potentially constitute copyright infringement of the software owner's exclusive rights under copyright law.

In addition to granting rights and imposing restrictions on the use of software, software licenses typically contain provisions which allocate liability and responsibility between the parties entering into the license agreement. Software licenses can generally be fit into the following categories: proprietary licenses and free and open source. The significant feature that distinguishes them are the terms under which the end-user may further distribute or copy the software.

GNU

The GNU Project is an effort of several programmers and developers to create a free, Unix-like operating system. GNU is a recursive acronym that stands for GNU is Not Unix, because it is Unix-like but contains no Unix code and is (and remains) free. The GNU foundation, the legal entity behind the GNU project, sees free as more than just the financial meaning of free: the

software should be free to use for any purpose whatsoever, free to study and modify the source code and behaviour, free to copy and free to distribute the changes you made.

This idea of free software is a noble thought that is active in many programmers' minds: hence many software titles are freely available. Software is generally accompanied by a license that explains what you can and cannot do with it (also known as the "End User License Agreement"). Free Software also has such a license - unlike the EULAs they actually allow most things instead of denying it. An example of such license is the GPL - GNU General Public License.

OSS or Open Source Software

The term "open source" refers to something that can be modified and shared because its design is publicly accessible.

While it originated in the context of computer software development, today the term "open source" designates a set of values—what we call the open source way. Open source projects, products, or initiatives are those that embrace and celebrate open exchange, collaborative participation, rapid prototyping, transparency, meritocracy, and community development.

OSS - Open source software is software whose source code is available for modification or enhancement by anyone. "Source code" is the part of software that most computer users don't ever see; it's the code computer programmers can manipulate to change how a piece of software—a "program" or "application"—works. Programmers who have access to a computer program's source code can improve that program by adding features to it or fixing parts that don't always work correctly.

OSS Difference with Others

Some software has source code that cannot be modified by anyone but the person, team, or organization who created it and maintains exclusive control over it. This kind of software is frequently called "proprietary software" or "closed source" software, because its source code is the property of its original authors, who are the only ones legally allowed to copy or modify it. Microsoft Word and Adobe Photoshop are examples of proprietary software. In order to use proprietary software, computer users must agree (usually by signing a license displayed the first time they run this software) that they will not do anything with the software that the software's authors have not expressly permitted.

Open source software is different. Its authors make its source code available to others who would like to view that code, copy it, learn from it, alter it, or share it. LibreOffice and the GNU Image Manipulation Program are examples of open source software. As they do with proprietary software, users must accept the terms of a license when they use open source software—but the legal terms of open source licenses differ dramatically from those of proprietary licenses. Open source software licenses promote collaboration and sharing because they allow other people to make modifications to source code and incorporate those changes into their own projects. Some open source licenses ensure that anyone who alters and then shares a program with others must also share that program's source code without charging a licensing fee for it. In other words, computer programmers can access, view, and modify open source software whenever they like—as long as they let others do the same when they share their work. In fact, they could be violating the terms of some open source licenses if they don't do this.

So as the Open Source Initiative explains, "open source doesn't just mean access to the source code." It means that anyone should be able to modify the source code to suit his or her needs, and that no one should prevent others from doing the same. The Initiative's definition of "open source" contains several other important provisions.

OSS Advantages

Open source software benefits programmers and non-programmers alike. In fact, because much of the Internet itself is built on many open source technologies—like the Linux operating system and the Apache Web server application—anyone using the Internet benefits from open source software. Every time computer users view webpages, check email, chat with friends, stream music online, or play multiplayer video games, their computers, mobile phones, or gaming consoles connect to a global network of computers that routes and transmits their data to the "local" devices they have in front of them.

OSS and Free

This is a common misconception about what "open source" implies. Programmers can charge money for the open source software they create or to which they contribute. But because most open source licenses require them to release their source code when they sell software to others, many open source software programmers find that charging users money for software services and support (rather than for the software itself) is more lucrative. This way, their software remains free of charge and they make money helping others install, use, and troubleshoot it.

1.2. Distributions and Linux evolution

A typical Linux distribution comprises a Linux kernel, GNU tools and libraries, additional software, documentation, a window system (the most common being the X Window System), a window manager, and a desktop environment. Most of the included software is free and open-source software made available both as compiled binaries and in source code form, allowing modifications to the original software.

Linux Distributions

A distribution is a collection of software (called the packages) bundled together in a coherent set that creates a fully functional environment. The packages contain software titles (build by other projects) and possibly patches (updates) specific for the distribution so that the package integrates better with other packages or blends in better with the overall environment. These packages are usually not just copies of the releases made by the other software projects but contain a lot of logic to fit the software in the global vision of the distribution.

Take KDE for example. KDE is a (graphical) desktop environment which bundles several dozens of smaller tools together. Some distributions provide a pristine KDE installation to their users, others change KDE a bit so that it has a different default look and such.

Another example would be MPlayer, a multimedia player especially known for its broad support of various video formats. However, if you want to view Windows Media Video files (WMV), you need to build in support for the (non-free) win32 codecs. Some distributions provide MPlayer with

support for these codecs, others without. Gentoo Linux lets you choose if you want this support or not.

Distributions are maintained by private individuals and commercial entities. A distribution can be installed using a CD that contains distribution-specific software for initial system installation and configuration. For the users, most popular distributions offer mature application management systems that allow users to search, find, and install new applications with just a few clicks of the mouse.

There are, at last count, over 350 distinct distributions of Linux. Various distributions will allow:

- ✓ Different tools and methods to set the system up.
- ✓ Different choices with regard to setup.
- ✓ Various choices of packages to install.

The core of the operating system is still likely to be the same or similar and many of the packages used will be the same. Also the user can get additional packages and install them on the operating system.

The various packages are created by different groups of developers on various computers and tested individually under different circumstances. There is a pretty good consensus of programming methodology among the Linux community with regard to program compatibility. There are various libraries that are used for compiling programs, and there are libraries providing various functions that are considered to be the "standard of the day". Although these issues are dealt with, there is no guarantee that all packages will work well together. Therefore some system testing should be done before a distribution is made available. The amount of testing varies widely from one distribution to another.

Distribution Differences - There are multiple different Linux distributions. Many have different philosophies – some, like Fedora, refuse to include closed-source software, while others, like Mint, include closed-source stuff to make it easier on users. They include different default software – like how Ubuntu includes Unity, Ubuntu derivatives include other desktop environments, Fedora includes GNOME Shell, and Mint includes Cinnamon or MATE.

Many also use different package managers, configuration utilities, and other software. Some distributions are bleeding edge and won't receive support for very long. Others, such as Ubuntu LTS or Red Hat Enterprise Linux, are designed to be stable distributions that will be supported with security updates and bug fixes for many years.

Some Linux distributions are intended for desktop computers, some for servers without a graphical interface, and others for special uses, such as home theater PCs. Some are designed to work out of the box – like Ubuntu – while others require a bit more tweaking, such as Arch Linux.

Selecting Distribution - Different Linux distributions are suited for different purposes. Which Linux distribution you should choose will depend on what you're doing with it and your personal preferences.

If you're a desktop user, you'll probably want something simple, like Ubuntu or Mint. Some people may prefer Fedora, openSUSE, or Mageia (based on Mandriva Linux). People looking for a more stable, well-tested system may want to go with Debian, CentOS (a free version of Red Hat Enterprise Linux), or even Ubuntu LTS.

There's no one right distribution for everyone, although everyone has a favorite. Linux distributions offer choice, which can be messy, but also very useful. Anyone can make their own distribution by assembling it from the source code themselves, or even taking an existing distribution and modifying it – that's why there are so many Linux distributions.

Linux Evolution

On August 25, 1991, a Finn computer science student named Linus Torvalds made the following announcement to the Usenet group comp.os.minux:

"I'm doing a (free) operating system (just a hobby, won't be big and professional like gnu) for 386(486) AT clones. This has been brewing since april, and is starting to get ready. I'd like any feedback on things people like/dislike in minix, as my OS resembles it somewhat (same physical layout of the file-system (due to practical reasons) (among other things)."

The "Minix" Torvalds referred to as a variant of the UNIX operating system, used as a guideline for his the free operating system he wanted to run on the x86-based consumer PCs of the day. "gnu" refers to the set of GNU (GNU Is Not Unix) tools first put together by Richard Stallman in 1983. UNIX, the operating system that started it all, had its origins in the old Bell Labs back in the early 60s.

Torvalds built the core of the Linux operating system, known as the kernel. A kernel alone does not make an operating system, but Stallman's GNU tools were from a project to create an operating system as well--a project that was missing a kernel to make Stallman's operating system complete. Torvalds' matching of GNU tools with the Linux kernel marked the beginning of the Linux operating system as it is known today.

Linux is in many ways still only at the beginning of its potential, even though it has enjoyed tremendous success since Torvalds' first request for help in 1991.

Linux has gained strong popularity amongst UNIX developers, who like it for its portability to many platforms, its similarity to UNIX, and its free software license. Around the turn of the century, several commercial developers began to distribute Linux, including VA Linux, TurboLinux, Mandrakelinux, Red Hat, and SuSE GMbH. IBM's 2000 decision to invest $2 billion in Linux development and sales was a significant positive event to the growth of Linux.


Certifications
Accounting, Banking and Finance
– Certified AML-KYC Compliance Officer
– Certified Business Accountant
– Certified Commercial Banker
– Certified Foreign Exchange Professional
– Certified GAAP Accounting Standards Professional
– Certified Financial Risk Management Professional
– Certified Merger and Acquisition Analyst
– Certified Tally 9.0 Professional
– Certified Treasury Market Professional
– Certified Wealth Manager
Big Data
– Certified Hadoop and Mapreduce Professional
Cloud Computing
– Certified Cloud Computing Professional
Design
– Certified Interior Designer
Digital Media
– Certified Social Media Marketing Professional
– Certified Inbound Marketing Professional
– Certified Digital Marketing Master
Foreign Trade
– Certified Export Import (Foreign Trade) Professional
Health, Nutrition and Well Being
– Certified Fitness Instructor
Hospitality
– Certified Restaurant Team Member (Hospitality)
Human Resources
– Certified HR Compensation Manager
– Certified HR Stafffing Manager
– Certified Human Resources Manager
– Certified Performance Appraisal Manager
Office Skills
– Certified Data Entry Operator
– Certified Office Administrator
Project Management
– Certified Project Management Professional
Real Estate
– Certified Real Estate Consultant
Marketing
– Certified Marketing Manager
Quality
– Certified Six Sigma Green Belt Professional
– Certified Six Sigma Black Belt Professional
– Certified TQM Professional
Logistics & Supply Chain Management
– Certified International Logistics Professional
– Certified Logistics & SCM Professional
– Certified Purchase Manager
– Certified Supply Chain Management Professional
Legal
– Certified IPR & Legal Manager
– Certified Labour Law Analyst
– Certified Business Law Analyst
– Certified Corporate Law Analyst
Information Technology
– Certified ASP.NET Programmer
– Certified Basic Network Support Professional
– Certified Business Intelligence Professional
– Certified Core Java Developer
– Certified E-commerce Professional
– Certified IT Support Professional
– Certified PHP Professional
– Certified Selenium Professional
– Certified SEO Professional
– Certified Software Quality Assurance Professional
Mobile Application Development
– Certified Android Apps Developer
– Certified iPhone Apps Developer
Security
– Certified Ethical Hacking and Security Professional
– Certified Network Security Professional
Management
– Certified Corporate Goverance Professional
– Certified Corporate Social Responsibility Professional
Life Skills
– Certified Business Communication Specialist
– Certified Public Relations Officer
Media
– Certified Advertising Manager
– Certified Advertising Sales Professional
Sales, BPO
– Certified Sales Manager
– Certified Telesales Executive
& many more job related certifications
Contact us at :
Vskills
011-473 44 723 or info@vskills.in
www.vskills.com