SECTION TWO – KNOWLEDGE PROFILE

Element One

A systematic, theory-based understanding of the natural sciences applicable to your discipline (e.g. calculus-based physics)

Context

All engineering fields are rooted in one or more of the natural sciences. In a broad context, natural science is separated into physical and biological sciences. Physical sciences include chemistry, calculus-based physics, astronomy, geology, geomorphology, and hydrology. Biological sciences involve living systems and include biology, physiology, microbiology, and ecology.

Washington Accord graduates are expected to be able to apply this knowledge of the natural sciences to solve complex engineering problems in their discipline.

Performance Indicators

- Fundamental quantitative knowledge underpinning nature and its phenomena.
- Knowledge of the physical world including physics, chemistry and other areas of physical or biological science relevant to your discipline
- Knowledge of key concepts of the scientific method and other inquiry and problem-solving processes.
- Application of knowledge from one or more of the natural sciences to the solution of complex engineering problems relevant to your discipline.

Summarise your knowledge of the natural sciences relevant to your discipline and how it has been developed through formal study, on-job learning and/or continuing professional development. **Note: please cross-reference to your academic transcript(s) and continuing professional development records, as appropriate.**	**Provide annotations to your supplementary evidence (document and page number)**
During my bachelors programme I studied various subjects that contained one course from Applied Physics, and four courses from applied mathematics. In computer engineering's context, these subjects helped me to solve complex scientific problems during my studies and during my professional career. Subjects related to maths and physics helped me understand key concepts in computer science, electrical engineering, and electronic engineering. For example, physics helped me understand how data is written, on CDs and how optics (a very important concept in Physics) is related to Laser technology. CD technology is an excellent example of the application of basic principles of physics, from the derivation of the signal by Interference to the dependence of audio frequency response and playing time on the optical wavelength used. Similarly, we need Maths and Physics in computer engineering when we dive into topics like the mathematics behind light, electricity, magnetism, AC/DC circuits, RF theory, Solid-state devices, Satellite Communications, and digital electronics. Beside above subjects I also studied Basic Electronics, Electric Circuit and Machines, Circuit Analysis, Electronic Instrumentations, Logic Designing and Switching Theory I and II. All these subjects are directly or indirectly rooted from applied physics.	

Applied Physics:

In applied physics, I learned how physics is particularly useful for technological and practical purpose in engineering. I studied properties of Matter that poses elasticity, moduli of Elasticity, Experimental determination of Young's modulus, Bending of beams, Cantilever. Young Modulus helped me to understand the relation between stress and strain in a material.

I also studied **Fluids, turbulent flow, and Bernoulli's theorem.** I came to know that the Bernoulli theorem has a wide range of applications in engineering fluid dynamics, from aerospace to hydroelectric plants and even in our everyday life when we turn on the ignition in vehicles that use Carburetor for engines.

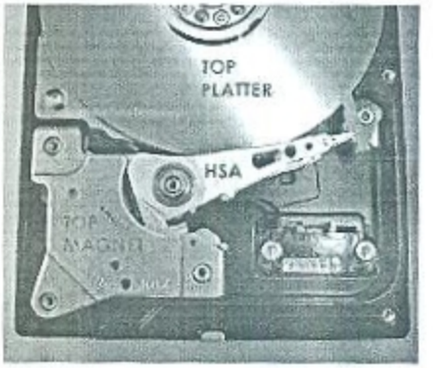
While studying **Viscosity** I learned how to determine coefficient of viscosity using Hagen Poiseuille law. I studied surface tension, surface energy, angle of contact, determination of surface tension by rise in a capillary tube. This law is useful in medical application to determine the flow rate of Intravenous fluids that may be achieved using various sizes of peripheral and central cannulas.

In **Heat &Thermodynamics** in applied physics I studied Heat, Temperature, and Theories of heat, Adiabatic and isothermal processes and also understood four laws of thermodynamics. In four laws I studied basic concepts of fundamental physical quantities (temperature, energy, and entropy) that characterize thermodynamic systems at thermal equilibrium and came to know that how these quantities behave under various circumstances. I also studied Thermodynamic functions, Maxwell's Thermodynamic relations, Efficiency of Heat Engines, Carnot's Cycle, Entropy.

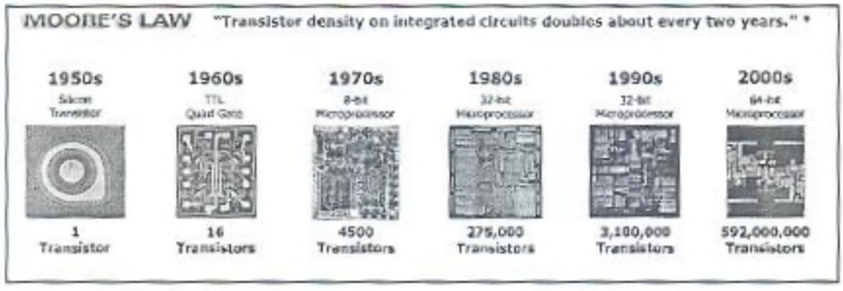
While studying **Optics** in Applied physics I learned Waves and Oscillations, Simple Harmonic Motion, types of wave motion, theories of light, Interference, Diffraction, Polarization, Double refraction, Dispersion, Deviation Lasers its type and uses.

I also studied **Electricity and Magnetism** in applied physics that helped me to understand fields of electric circuits and magnetic force. In this topic Coulomb's law helped me to understand electric charges, electric field, electrostatic potential of discrete charges and continuous charge distribution. In this topic Gauss's Law helped me to develop understanding about electric field around conductors, dielectrics, capacitors, electric current. I studied Ohm's Law, Magnetic field, Magnetic force on current, Ampere's law, Faraday's law, and Lenz's law. (Current A.C. and D.C. behaviour of RC, LR &LC in the presence of DC field and LRC in the presence of AC field).

A good example of using magnetism in our everyday life is usage of Hard Disks for data storage. These days all IT systems require data storage devices and over the years this technology has greatly evolved into one of the core computer system areas.



Study about transistors which is another core concept in computer engineering. I learned about semiconductor physics and also studied how conduction occurs in a metal. I understood concepts like acceptors, holes, N & P type doped and compensated semiconductors, Junctions, forward and reverse bias, diode action as P-N Junction, Transistor and its characteristics. The transistor is the fundamental building block of modern electronic devices, mainly radios, calculators and computers. It will not be wrong to say that our current world is actually built upon the transistor technology. Below image shows the evolution of transistor technology over last few decades.



Applied Mathematics

In Applied Mathematics I studied four subjects

a) Applied Mathematics I:

BS(Computer Engineering) Transcript

Course: MS102

This course contains thorough studies of Linear Algebra, Coordinate Geometry of 2-Dimensions.

In **Linear Algebra**, I studied Complex numbers and their properties, geometric representation of a complex number. Formation and solution of quadratic equations, cube roots of unity. Arithmetic, geometric and harmonic progressions, infinite geometric series, permutations and combinations, probability. Binomial expansions for positive integral exponents, general and middle terms or terms of the expansions. Partial fractions.

In **Coordinate Geometry of 2-Dimensions** I studied Functions and their types. Cartesian and polar coordinates, distance formula, midpoint of a line segment, Equations of a line, angle between lines, distance of a point from a line.

b) Applied Mathematics II:

BS(Computer Engineering) Transcript

Course: MS103

This course contains thorough studies of Determinants and Matrices, Trigonometry, Differential Calculus, Integral Calculus.

In **Determinants and Matrices**, I studied their properties, matrix operations, inverse of a matrix, Cramer's rule.

In **Trigonometry** I studied Circular measure, relationship between radians and degrees. Trigonometric functions and their identities, sum and difference formulae, multiple-angle formulae, Inverse trigonometric functions.

In **Differential Calculus** I studied Limits of elementary functions. Differentiation of elementary functions. Sum, product and quotient formulae. Differentiation for implicit and parametric functions. Higher derivatives, Leibniz formula. Maxima and minima, tangents and normal's, velocity and acceleration, rate of reaction etc.

In **Integral Calculus**, I studied Integration of elementary functions. Integration by substitution, by partial fractions and by parts, definite integrals. Trapezoidal and Simpson's rules.

c) Applied Mathematics III

BS(Computer Engineering) Transcript

Course: MS201

This course contains thorough studies of Complex variable and integral transform, Partial Differentiation, Technique of Integration, Vector Algebra, Functions and Their types, Calculus and Analytical Geometry.

In **Complex Numbers** I studied Basic operations, graphical representations, polar and exponential forms of complex numbers, De Moivre's theorem with applications. Hyperbolic functions, exponential functions and their relationships in trigonometric identities. Review of differentiation, successive differentiation, Leibniz theorem, applications, equations of tangents and normal's, curvature, radius and centre of curvature, maxima and minima of functions, applications, curve tracing in rectangular and polar coordinates. I also studied Definition of **Laplace transform**, Laplace transforms of elementary functions, properties of Laplace transform, s-shifting, unit step functions, t-shifting, Dirac delta function, Laplace transform of derivatives and integrals, inverse Laplace transform and its properties, convolution theorem, solutions of ordinary differential equations by Laplace transforms.

In **Partial differentiation**, I studied partial differential coefficients, chain rule, differentiation of implicit functions, total differential, Euler's theorem, applications to small errors and approximations, Taylor's theorem of one and two independent variables (without proof) and applications.

In **Integration techniques** I studied integration by substitution, by parts and by partial fractions, integration of trigonometric functions, Riemann sums and definite integrals, fundamental theorem of integral calculus, improper integrals. Gamma and beta functions, applications of integration. Limits & continuity of several variables, partial derivatives, chain rule, Taylor's theorem for functions of two variables (without proof), maxima and minima of functions of two variables. Lagrange's multipliers. Jacobians, double integrals, change of order, conversion to polar form, applications.

In **Vector algebra**, I studied Differentiation and integration of vector functions, gradient, divergence and curl. Line integrals, Green's theorem, Gauss' divergence theorem. Rectangular, spherical and cylindrical coordinate systems in three dimensions, direction cosines, plane ,straight line and sphere.

In **calculus and Analytical** geometry, I studied Functions and their types, left hand and right hand limits, continuous functions. Differentiability, applications of derivatives, Rolle's theorem, mean value theorems, maxima and minima of a function of a single variable (applied problems), differentials with applications, asymptotes, tangents and normals, curvature and radius of curvature, curve sketching. Evaluation of limits using L' Hospital's rule.

d) Applied Mathematics IV

BS(Computer Engineering) Transcript

Course: MS202

This course contains thorough studies of Differential Equations, Periodic functions and expansion of periodic functions in Fourier series, Partial Differential Equations, and Discrete Mathematics.

In **differential equation,** I studied its classification, formation and solution of differential equations, initial and boundary conditions, methods of solution of ordinary differential equations of the first order and first degree, separable, homogeneous, reducible to homogeneous and exact differential equations, integrating factors, linear equations, Bernoulli equations, orthogonal trajectories in Cartesian and polar coordinates.

In **Periodic functions and expansion of periodic functions in Fourier series**, I studied expansion of functions with arbitrary periods. Odd and even functions and their Fourier series, half-range expansions. Basic concepts and formation of partial differential equations, linear homogeneous partial differential equations and their relations to ordinary

differential equations, solution of first order linear partial differential equations, two dimensional wave equations. D' Alembert's solution of the wave equation.

In **Partial Differential Equations,** I studied Basic concepts and formation of partial differential equations, linear homogeneous partial differential equations and their relations to ordinary differential equations, solution of first order linear partial differential equations, two dimensional wave equations, D' Alembert's solution of the wave equation.

In **discrete mathematics**, I studied Propositional logic in which I further studied sets, principle of inclusion-exclusion, functions, pigeonhole principle, relations and their properties, representing relations, closure, equivalence relations, equivalence classes, partial ordering, methods of proof and mathematical induction. Graphs and digraphs, representation of graphs, paths and connectivity, planar graphs. Weighted and acyclic digraphs. Integers, divisibility, primes, relative primes, prime factorization, fundamental theorem of arithmetic, Euclid's division algorithm, GCD, LCM, modular arithmetic and congruence. Binary operations, semi groups and monoids. Groups, subgroups, rings and fields.

Examples:

Linear Algebra made me understand and solve various resistive circuits' issues in Circuit Analysis Subject. While studying **Boolean algebra**, I studied that it is core of various digital logic gates and switches and it helped me to develop circuit design and to solve relevant complex problems in logic designing and switch theory subject. **Matrices** helped to understand and solve various complex electric circuits and their derivative in Basic Electrical Circuit subject.

Scalars and Vectors, the Cartesian coordinate system (coordinated geometry of 2 dimensions), **Vector algebra** helped me to developed understanding about electromagnetic fields. In **Integral calculus (Differentiation)**, I came to know how displacement from velocity and velocity from acceleration is calculated, I also learned to calculate the voltage across capacitor while studying circuit analysis course and other applications. **Integration** helped me to understand how volume of objects can be calculated having circular cross sections and curved sides while studying computer graphic and animation subject.

In **Differential Equations (Linear)** I studied modelling of electrical circuits i.e. LR Circuits and RC Circuits while studying circuit analysis

Discrete mathematics developed reasoning ability in myself and make me able to understand mathematical statements. Studying this subject I came to know about the science of digital computers that work with discrete steps and store data in discrete bits. This subject has added further value to my studies in computer algorithms, programming language, cryptography. In discrete mathematics, one most important topic is Propositional logic that make me understand Logic Gates and programming syntax, and helped me to solve complex logic in computer programming. Propositional Logic is concerned with statements to which the truth values, "true" and "false", can be assigned. The purpose is to analyze these statements either individually or in a composite manner.

A proposition is a collection of declarative statements that has either a truth value "true" or a truth value "false".

Examples of Propositions are as follows

- "12 + 9 = 3 – 2", it returns truth value "FALSE"
- There are statements in math like "10-4=6" and "1+1=3", in this one of the result is true and one is false

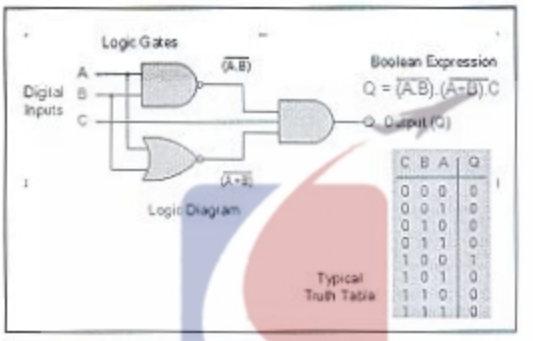
In propositional logic generally we use five connectives which are - OR (∨), AND (∧), Negation/ NOT (¬), Implication / if-then (→), If and only if (⇔).

INPUTS		OUTPUTS					
A	B	AND	NAND	OR	NOR	EXOR	EXNOR
0	0	0	1	0	1	0	1
0	1	0	1	1	0	1	0
1	0	0	1	1	0	1	0
1	1	1	0	1	0	0	1

Propositional Variable and Connections

Based on the studies of **Functions** and their types I was able to understand that mathematical functions are the central objects of investigations, it has helped me to understand how a function can compute the output for a given input. Using mathematical functions I was able to analyze various graphs with various dimension having outputs with different combination of inputs.

I studied that **Binary system,** that is the core of digital system in computer engineering that helped me to understand how digital data is handled in binary terms. I learned that binary numbers are the roots of digital and computer circuits. They are represented by either logic "0" or a logic "1" and computer exchanges and processes information in the ones and zeros of binary. This subject has helped me to understand the basic and advance computer architecture and organization based on binary calculation.



I studied that **Laplace transform** is a large class of function. It helped me to understand series RLC circuit while studying circuit analysis and its application in Power Systems Load Frequency control. This function has also helped me to understand different areas of physics and electrical power engineering **complex numbers and complex number systems** helped me to develop understanding of Argand diagram, curve and regions in the complex plane, root of polynomial equations, standard functions and their inverses.

Basic Electronics & Electric Circuit and Machines

In these two subjects I studied **The physical foundation of Electric Circuits**: Structure of Matter, Conductors, Insulators and Semiconductor, Electric Current, Electromotive force (voltage), Resistance, Conventional Current, DC and AC, Ohm's Law, Work, Energy and Power, Conductance, Efficiency, Real and Ideal Sources, Circuit Analysis, **Resistive Network**: Kirchoff's Voltage and Current Laws, The Voltage-Divider Rule, The current Divider Rule, Series and Parallel Connected Sources, Wye-Delta transformations, Balance Bridges, Voltage and Current Source Conversions, Mesh Analysis, Nodal Analysis, **Network theorems:** The superposition theorem, thevenin's theorem, Norton's theorem and maximum power transfer theorem. **Capacitance and Capacitors:** The nature of Capacitance, Capacitor Dimensions and Dielectrics, capacitor types and ratings, transients in RC networks, Energy stored in a capacitor. **Inductance and Inductors**: Electromagnetic Induction, Lenz's law, Faraday's law, Transformer Action, Self Inductance, Inductors, Transients in RL Circuits, Energy stored in an inductor. **Alternating Current Fundamentals:** AC Wave-form, Period and Frequency, Radians and Angular Frequency,

BS(Computer Engineering) Transcript

Course: EE107 & EE213

Peak and Instantaneous values, Average and Effective values, AC voltage and current in capacitors and inductors, Average Power. I also Studied **Transformers**: Principal of operation, Construction, types, EMF equation and transformation ratio, equivalent values and equivalent power circuit diagram, impedance matching, operation and phasor diagram with and without load, measurement of losses and efficiency, parallel operation, cooling, three phase connections, instrumentation transformers **DC Machine**: Introduction to DC machines, Electromagnetic conversions, motoring and generating action, parts of DC machines and their working, types of DC Generators and DC motors, types of windings and EMF equation of DC generator, losses in DC generator **AC Machines**: Introduction to Induction machine, AC generator, Motor **Method of speed control of induction motor:** Voltage control, Frequency control, V/F control. Principals of three phase induction motor, Run Capacitor induction motor, Shaded stator pole induction motor, stepper motors, induction to synchronous motors.	
Circuit Analysis In this course I studied, **The RLC Circuit:** the source free series and Parallel RLC Circuit, the over damped, under damped, and critically damped RLC circuits, the complete response of RLC Circuit, the lossless LC circuit. **The sinusoidal steady state response**: Nodal, Mesh and loop analysis, AC source transformation, Thevenin's Norton's reciprocity and compensation theorems. **Complex Frequency**: Introduction to complex frequency, damped sinusoidal forcing function, Z(s) & Y(s), frequency response as a function of s, Complex frequency plane, natural response and the S-Plane, Synthesizing the voltage ratio H(s) - vout Vin. Scaling & Bode Diagrams. **General two port networks**: Introduction, admittance parameters, some equivalent networks, impedance parameters, hybrid parameters, transmission parameters. **Fourier Analysis & Fourier Transforms**: Trigonometric form of Fourier Series, the use of symmetry, Complete Response to periodic forcing functions, Complex form of Fourier series. Definition of Fourier Transform, some properties of Fourier Transform, the Unit-impulse Function, Fourier transform of a General Period Time function, Convolution and Circuit Response in the Time domain, the system function and response in the frequency domain, the physical significance of the system function. **Laplace Transform** Techniques: Definition of Laplace Transform, Convolution Time-Shift and Periodic Functions. Shifting, Differentiation, Integration and Scaling in the frequency Domain, The Transfer function (hs).	BS(Computer Engineering) Transcript Course: EE211
Electronic Instrumentations: In this course, I studied Introduction to Electronic Instrumentation, Definition and Purpose of Instrumentation, Data Analysis, Definition of Data, Precision, Accuracy , Data Presentation, Tabular and Graphing, Errors, Basic Definition, Types of Errors, Estimation of Errors, Functional Elements of a Typical Instrument Block Diagram, Model Relating Input and output Correction Methods, Static Evaluation, Calibration Technique, Evaluation of Static Parameters Sensitivity, Linearity, Resolution, Threshold, Hysterises, Dead Span, Rang, Basic Definition, Sources of Noise, Methods of Noise Coupling, Conductive, Capacitive, Magnetic and Electromagnetic, Noise Protection Methods, Shielding, Grounding, Safety Grounding, Signal Grounding, Noise in Digital Systems, Voltage Measurement Permanent Magnet Moving Coil (PMMC), Digital Voltmeters, Analogue Voltmeters, Increasing the range of PMMC (shunt resistors and series resistors), Current Measurement, Methods of Current Measurement, Resistance measurement, Methods of resistance measurement, Frequency and Phase measurement, Methods of frequency measurement, methods of phase measurement, Power and Power Quality Measurements, Oscilloscopes, Chart Recorders, Spectrum Analyzers & Signal Generation, Oscilloscope block diagram, cathode ray tube (CRT), CRT Circuits, Vertical deflection system, probes and transducers, digital storage scopes, types of chart recorders, introduction to spectrum analyzers, sine wave generator, signal generation modulation, sweep frequency generator, pulse and square wave generator, function generator, audio frequency generator, signal generator.	BS(Computer Engineering) Transcript Course: EE207

Logic Designing and Switching Theory I and II:

In this course I studied **Digital Systems:** The Binary System, Boolean Algebra and Reduction techniques with Boolean postulates, K-map and **Queine McCluskey** methods. Analysis and Design of Sequential Circuits, Combinational circuits and sequential circuits i.e. Multiplexer, de-multiplexer, decoder, ROM, PLA, PAL, half adder and full adder, Flip flops, Registers and Counter Circuits. **Introductory to Digital Concepts:** Digital and Analog quantities, Binary Digits, Logic Levels and Digital waveforms, Digital Logic Operations, basic logic, Digital System applications, Digital ICs, Classification of ICs, Integrated Technologies, Number systems and conversions, Logic Gates, Basic Logic Gates, Universal Logic Gates, The XOR, The Universal property of NAND & NOR Gates, Boolean Laws and Rules, Standard forms of Boolean expressions, Simplification of expression using Boolean algebra, De Morgan's Theorems, Simplification of Boolean Expressions using De Morgan's Theorems, SOP to POS and vice versa, The Kamaugh's map, 2-4 variable karnaugh's Map, Basic Adder circuits, Comparators, Encoders, Decoders, Multiplexers, De-Multiplexers, Flip-Flops and related devices, Latches and its types, Flip-Flop operating characteristics, Shift Registers, basic shift registers functions, Serial-in/Serial-out Shift registers, Serial-in/Parallel-out Shift Registers, Parallel-in/Serial out Shift Registers.

BS(Computer Engineering) Transcript

Course: EE212 & EE214

Basic Computer Design

In this course I studied Overview of Microprocessor Structure & operation, The 8086 Microprocessor Family/Overview, Review of 8086 interfacing to memory and IO, Interrupts and Interrupts Application, DA Converter Operation, Interfacing and Applications, AD Converter Specifications Types and Interfacing, introduction to microcontroller, Assembly language of a typical microcontroller, program control instruction, I/O Port Programming, serial Port Programming, ADC/DAC interfacing.

BS(Computer Engineering) Transcript

Course: CE307

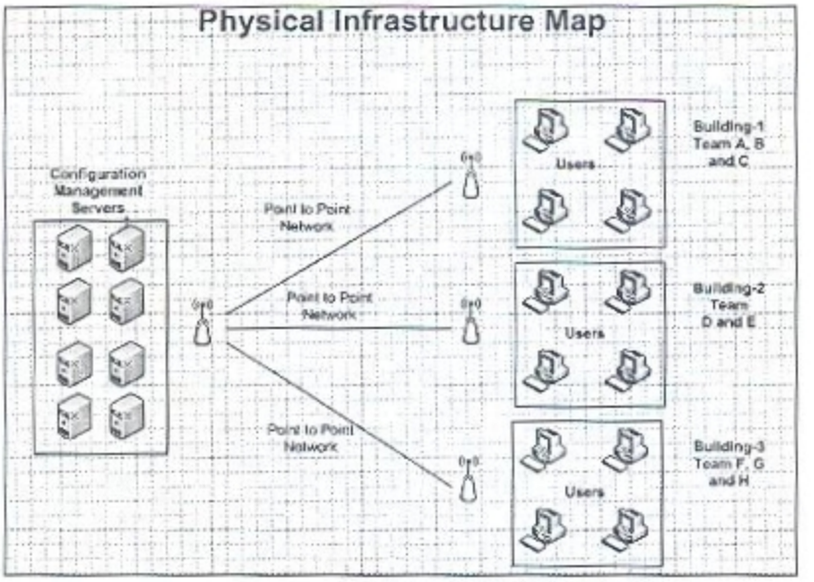
Example: Root Cause Analysis (RCA) and Rectification of Servers Performance due to Power Supply Outage: Period(2005-2006)

I worked in Kalsoft (Pvt) Ltd as Software Configuration Management Engineer and I was responsible for Managing and Administrating Configuration Management Servers. This was my first professional employment and it gave exposure to real life problems where I learned to solve complex engineering problems.

There were eight configuration Management servers setup in Microsoft Visual Source Safe 2005. The servers were located in one server room within same building where I was sitting. Servers were used by eight different teams located in different buildings. The users were connecting to servers via tower of point to point network. The repository was used by Software Engineering team for their code merging and retrieving of code data making connections to server from their local system as client. The repository was also being used by Customer Supplier team who were taking care of the customer related data in repositories.

The repository comprised of Entire Organizational and Projects Artefacts as follows

- Organizational Artifacts (Process Documents, Guidelines Documents, Training Documentations, Audit Reports, Customer Data etc)
- Project Artefacts (Project Plans, Source Code, Technical Documents etc)



Problem Statement:

While accessing configuration management serves all the users of the repositories were facing problems related to server connections and delays in merging and retrieving of codes.

My manager asked me to do Root Cause Analysis (RCA) and rectify the problem.

Findings:

When I engaged myself in this assignment, I started to analyze the problems and identified following observations.

- Data base size of each repository was increasing rapidly due to newly induct projects and code merging.
- Number of users connections to servers were increasing day by day.
- The server configurations like primary memory i.e. RAM and Secondary Memory i.e. Hard Drive capacity and processing power i.e. Processor was not enough to handle to overall performance due to which delays in code merging and making connection were occurring.
- The office building was using self generating power using Generator that start 8:00 Am in the morning and was schedule to stopped at 7:00PM in evening. After that all the servers were shifted to UPS power till morning.
- On daily basis all eight configuration management servers were getting rebooted whenever I came in the morning and login to system to perform my job.
- When I saw the system log, it was showing that data was being corrupted due to improper shutdown and improper reboot of servers.

I informed to Power management team to check the backup power of servers and they were repeatedly replying that Power was properly being supplied and servers have some problem that I need to check. At that time it was very challenging to work with power management team and make them work to identify the cause of power outage. After few days, I found that servers performance was degrading as I was receiving complaints from users . I raise this concern to management with all the logs details and insisted to make

power management team to check servers power backup in server room.

When I visited server room with power management team I found following observations.

- The Configuration Management Servers were plugged to UPS that have only capacity to backup eight servers, but I found that there were additional four other servers were also plugged with same UPS.
- The UPS management log was showing that batteries of UPS never got fully charged after it was first plugged. Here Electronic Instrumentation course has helped me to analyze this observation.
- UPS Self test log was not showing that when it was last verified for self check that its load doesn't exceed the maximum capacity.

Observations:

It was found that battery was being completely utilized till 11PM due to over utilization by additional servers and because of this reason servers were regularly powered off. The improper shutting down of servers were also causing Source Safe files crashed and degrading the server performance. Absence of regular monitoring of servers power backup and its self-test was also the main cause.

Steps for Resolution:

After finding the root cause, I took following steps.

1. Inform to all the users to disconnect from database and save their data in local working folders.
2. I make power management team to remove the additional servers from UPS making their separate arrangement and ensured that UPS batteries are fully charged and checked its health by self-test that it do not exceed the maximum capacity.
3. Taken backup of all eight servers repository databases on tap drives and verified its recovery.
4. I setup new eight servers with high configuration considering current and future need.
5. I restored the repository databases on new servers
6. After verification of connections, I intimated to all user to resume their work.

Element Two

Conceptually-based mathematics, numerical analysis, statistics and formal aspects of computer and information science to support analysis and modelling applicable to your discipline

Context

Branches of mathematics applied in engineering include arithmetic, algebra, geometry, trigonometry, calculus, differential equations, numerical analysis, optimization, probability and statistics, simulation, and matrix theory. Engineers apply mathematics in a wide variety of functions typically carried out in engineering organisations such as planning, design, manufacturing, construction, operations, finance, budgeting, and accounting.

Washington Accord graduates are expected to be able to apply this mathematical knowledge to solve complex engineering problems in their discipline.

Performance Indicators

- Knowledge of mathematics, statistics and numerical methods that supports the development or application of models that replicate 'real world' behaviours
- An understanding of the assumptions behind theoretical models and their impacts in the development and use of those models
- Ability to organise and analyse a data set to determine its statistical variability,
- Knowledge of trigonometry, probability and statistics, differential and integral calculus, and multivariate calculus that supports the solving of complex engineering problems.
- Ability to apply differential equations to characterize time-dependent physical processes

Summarise your mathematical knowledge relevant to your discipline and how it has been developed through formal study, on-job learning and/or continuing professional development.

Note: please cross reference to your academic transcript(s) and continuing professional development records, as appropriate.

Provide annotations to your supplementary evidence (document and page number)

During my bachelors programme I studied four courses from applied mathematics for which detailed description is provided in Element-1

During my bachelors programme I also studied Probability and Statistics having various important topics.

<u>Probability and Statistics:</u>

In this subject, I studied about **Descriptive Statistics,** it is the term given to the analysis of data that helps summarize data in a meaningful way. In this topic I studied about Basic definitions, measures of central tendency and variation, Chebychev's theorem, z-scores, frequency distribution, graphical representation of data, stem and leaf and box plots, symmetry and skewness, quintiles (percentiles, deciles and quartiles).

BS(Computer Engineering) Transcript

Course: MS301

I also studied about **Probability theory,** that is all about to deals with quantities having random distributions. In this topic I studied basic definitions and rules of probability, conditional probability and Bayes' theorem, counting techniques.

I studied the topic of **Random variable.** It is usually written as X, as a variable whose possible values are numerical outcomes of a random phenomenon. In this topic I studied about the concept of random variable, discrete and continuous random variable and variance

of random variable and their properties.

I also studied the topic of **Discrete and continuous probability distributions.** This is actually a cumulative probability. In this topic I studied about Uniform, binomial, multinomial, hyper geometric, negative binomial, geometric, Poisson, normal and exponential distributions and their applications.

In this subject I studied about Sampling Theory, it is a field of statistics that involve collection, analysis and interpretation of data gathered from random samples of a population. In this topic, I studied about sampling distribution of mean, t-distribution and sampling procedures.

In this subject, I studied about **Regression and correlation**. Both Regression and Correlation are related analysis, in sense that both deal with relationships among variables. The correlation coefficient is a measure of linear association between two variables. Values of the correlation coefficient are always between -1 and +1. In this, I studied about Linear, exponential, multiple regression models, and multiple correlation coefficients.

I studied about **Statistical inference** in which I studied about estimation of parameters such as mean and variance, classical and Bayesian method of estimation.

I also studied about **Hypothesis testing** in which z-test, t-test and goodness of fit test were important topics.

I have been using mathematic during on job learning in various occasions. I have used metrics in measuring software-testing progress. Following are the examples of metrics I have used in order to measure and monitor projects progress and status.

Example 2-1: Calculating Variance

Variance is the metric to measure extra time spend in software testing against the planned effort. I calculate variance to see how much the effort is deviated from the planned effort.

Variance can be calculated by using mathematical formula as

Effort Variance (%) = (Actual Effort - Planned effort)/ Planned Effort) * 100

The effort variance may be greater than expected. For example, 100hrs were estimated for the testing project but the actual testing took 110hrs. In this case, if variance is calculated it will be

Effort Variance (%)= (110 - 100) / 100 = 10%

The effort variance may be less than expected. For example, 100hrs were estimated for test the project but the actual testing took 90hrs. Based on calculation, the result will be

Effort Variance (%) = (90 - 100) / 100 = -10%

Example 2-2: Calculating Requirements Volatility

Requirement volatility is a most common and crucial aspect of software development projects. Whenever changes occurred in software requirement due to any addition, deletion or modifications, requirements volatility is calculated in order to manage the change. It is measure in percentage.

For Example there were total 100 numbers of requirements. Later on 20 requirements were added, 10 deleted and 30 modified

Requirements volatility can be calculated by using mathematical formula as

Requirement Volatility (%) = [(No. of requirements added + No. of requirements deleted + No. of requirements modified) / No. of initial approved requirements] * 100

Requirement volatility (%) = (20+10+30)/100 = 60%

Most of the software development company create some internal standards to cost the project if the volatility in requirement occurred by customer. Like if volatility is between 30% to 40% then 10% of overall project cost will be additionally charged.

Example 2-3: Calculating Productivity of Test Execution

Test Execution productivity is the total number of test case executed divided by the total time it took for execution. It is normally calculated to measure the productivity of test case execution during testing phase in different period of time. It is calculated to know the status of software testing projects.

Test Execution Productivity can be calculated by using mathematical formula as

Productivity of Test Execution = (No of test cases executed / Time spent in test execution)

Let consider that unit of effort is measure as test cases per person per day and the time is the cumulative time of all the resources.

Let say, If there were 1000 test cases to be executed in a cycle by four software tester engineers.

Let say, First software tester executed 300 test cases in 2 days,
Second software tester executed 400 test cases in 3 days
Third software tester executed 75 test cases in 1 day
Fourth software tester executed 225 test cases in 4 days.

The cumulative time spent for executing 1000 test cases will be 10 man-days.

The Productivity of Test Execution will be calculated as

Test Execution Productivity = Total Test Cases Executed/Total time required by 4 testers

=1000/10 = 100

Therefore, the productivity in test execution is 100 test cases per person per day.

Example 2-4: Calculating Defect Rejection Ratio

It is calculated to assess the performance of software test engineer. In this metric, it is calculated that how much issues were reported that have no impact on software quality. It is considered as waste effort or an opportunity for improvement.

Defect Rejection ratio can be calculated by using mathematical formula as

Defect Rejection Ratio (%) = (No. of defects rejected / Total no. of defects raised) * 100

For example, there are total 150 issues were reported with different levels of severity and priorities, in which 80 issues were rejected.

Defect Rejection ratio will be calculated as = (80/150) * 100 = 53.3%

Many organization who work to improve their processes by following best practices, they

normally assess the performance of their software test engineer by using this formula and contribute the outcome in their annual appraisals. Some time due to criticality of product quality, if the defect rejection ratio show greater percentage (%), then project manager some time decide to place the more competent software test engineer to save time and to identify more qualitative issues.

Example 2-5: Calculating Defect Fix Rejection Ratio

Defect Fix Rejection ratio is the ratio between number of defects rejected and no of defects fixed. It is used to assess the performance of software developer who have fixed the issues but marked rejected by software test engineer due to incorrect fix.

It can be calculated by using mathematical formula as

Defect Fix Rejection Ratio (%) = (No. of defect fixes rejected / No. of defects fixed) * 100

For example, there are 50 issues were reported and developer fixed all the 50 issues, but when software tester verified, he found 30 issues are reopened due to incorrect fixes. In this case defect fix rejection ratio will be calculated as

Defect Fix Rejection Ratio = (30/50) * 100 = 60%

Example 2-6: Calculating Defect Removal Efficiency

The defect removal efficiency is the measurement to assess development team ability to remove defects before or at the moment of software product release. It is calculated as a ratio of defects resolved out of total number of defects found.

It can be calculated by using mathematical formula as

Defect Removal Efficiency (%)= Number of Defects resolved / Total Number of Defects at the time of Measurement.

For example, 100 defects are found after complete testing phase, out of which 84 defects were resolved.

The Defect Removal Efficiency will be = (84/100) X 100= 84%

Example 2-7: Calculating Bug Density

Bug Density is also called Defect Density. It is the number of defects confirmed in software or its module during a specific period of development divided by the size of the software or its module. This matrix helps to decided if the software is ready to be released or not.

It can be calculated by using mathematical formula as

Bug Density = Total bugs / Total Line of Code (Size)

For Example, there are four modules of the software product

Module 1 = Total bugs 20
Module 2 = Total bugs 30
Module 3 = Total bugs 50
Module 4 = Total bugs 60

Total Line of code for each module is

Module 1 = 1200 Line of Code (LOC)
Module 2 = 3023 Line of Code (Loc)

Module 3 = 5034 Line of Code (Loc)
Module 4 = 6032 Line of Code (Loc)

Total Bugs = 20 + 30 + 50 + 60 = 160

Size (Total Line of Code) = 1200 + 3023 + 5034 + 6032 = 15289

Bug density will be calculated as

Bug Density = 160 / 15289 = 0.010465

Bug Density = 0.010465 * 1000= 10.465 defects /KLOC

Example: S-Curve for Monitoring the Progress of Software Testing Showing theoretical and actual graph of Passed and Failed Tests:

In order to monitor the progress of software testing project, I have developed Test Execution Matrix as follows showing status of each test case executed.

Consider that there are 150 total number of test cases from which 70 have been attempted to execute.

Note: Following table showing 70 test cases status

No	TC ID	Run Date	Actual Results	Run Status	Current Status	No of Runs
1	SST-001	1-Jan-16	Test Case Passed	P	P	1
2	SST-002	1-Jan-16	Test Case Failed	F	F	1
3	SST-003	2-Jan-16	Test Case Failed	F F P	P	3
4	SST-004	2-Jan-16	Test Case Passed	P	P	1
5	SST-005	2-Jan-16	Test Case Passed	P	P	1
6	SST-006	2-Jan-16	Under Investigation	U	U	1
7	SST-007	2-Jan-16	Test Case Passed	P	P	1
8	SST-008		Blocked	B	B	0
9	SST-009		Blocked	B	B	0
10	SST-010	3-Jan-16	Test Case Passed	P	P	1
11	SST-011	3-Jan-16	Test Case Passed	P	P	1
12	SST-012	3-Jan-16	Test Case Passed	P	P	1
13	SST-013	3-Jan-16	Test Case Passed	P	P	1
14	SST-014	3-Jan-16	Test Case Passed	P	P	1
15	SST-015	3-Jan-16	Test Case Passed	P	P	1
16	SST-016	3-Jan-16	Test Case Passed	P	P	1
17	SST-017	3-Jan-16	Test Case Passed	P	P	1
18	SST-018	3-Jan-16	Test Case Passed	P	P	1
19	SST-019	3-Jan-16	Test Case Passed	P	P	1
20	SST-020	3-Jan-16	Test Case Passed	P	P	1
21	SST-021	3-Jan-16	Test Case Passed	P	P	1
22	SST-022	3-Jan-16	Test Case Passed	P	P	1
23	SST-023	3-Jan-16	Test Case Passed	P	P	1
24	SST-024	3-Jan-16	Test Case Passed	P	P	1
25	SST-025	3-Jan-16	Test Case Passed	P	P	1
26	SST-026	3-Jan-16	Test Case Passed	P	P	1
27	SST-027	4-Jan-16	Test Case Failed	F	F	1
28	SST-028	4-Jan-16	Test Case Failed	F	F	1
29	SST-029	4-Jan-16	Test Case Failed	F	F	1

30	SST-030	4-Jan-16	Test Case Failed	F							F	1	
31	SST-031	4-Jan-16	Test Case Passed	P							P	1	
32	SST-032	4-Jan-16	Test Case Passed	P							P	1	
33	SST-033	4-Jan-16	Test Case Passed	P							P	1	
34	SST-034	4-Jan-16	Test Case Passed	P							P	1	
35	SST-035	4-Jan-16	Test Case Passed	P							P	1	
36	SST-036	4-Jan-16	Test Case Passed	P							P	1	
37	SST-037	4-Jan-16	Test Case Passed	P							P	1	
38	SST-038	4-Jan-16	Test Case Passed	P							P	1	
39	SST-039	4-Jan-16	Test Case Failed	F							F	1	
40	SST-040	4-Jan-16	Test Case Failed	F							F	1	
41	SST-041	5-Jan-16	Test Case Passed	P							P	1	
42	SST-042	5-Jan-16	Test Case Passed	P							P	1	
43	SST-043	5-Jan-16	Test Case Passed	P							P	1	
44	SST-044	5-Jan-16	Test Case Passed	P							P	1	
45	SST-045	5-Jan-16	Test Case Passed	P							P	1	
46	SST-046	5-Jan-16	Test Case Failed	F							F	1	
47	SST-047	5-Jan-16	Test Case Failed	F							F	1	
48	SST-048	5-Jan-16	Test Case Failed	F							F	1	
49	SST-049	5-Jan-16	Test Case Failed	F							F	1	
50	SST-050	5-Jan-16	Test Case Failed	F							F	1	
51	SST-051	6-Jan-16	Test Case Passed	P							P	1	
52	SST-052	6-Jan-16	Test Case Passed	P							P	1	
53	SST-053	6-Jan-16	Test Case Passed	P							P	1	
54	SST-054	6-Jan-16	Test Case Passed	P							P	1	
55	SST-055	6-Jan-16	Test Case Passed	P							P	1	
56	SST-056	6-Jan-16	Test Case Failed	F							F	1	
57	SST-057	6-Jan-16	Test Case Failed	F							F	1	
58	SST-058	6-Jan-16	Test Case Failed	F							F	1	
59	SST-059	6-Jan-16	Test Case Failed	F							F	1	
60	SST-060	6-Jan-16	Test Case Failed	F							F	1	
61	SST-061	7-Jan-16	Test Case Passed	P							P	1	
62	SST-062	7-Jan-16	Test Case Passed	P							P	1	
63	SST-063	7-Jan-16	Test Case Passed	P							P	1	
64	SST-064	7-Jan-16	Test Case Passed	P							P	1	
65	SST-065	7-Jan-16	Test Case Passed	P							P	1	
66	SST-066	7-Jan-16	Test Case Failed	F							F	1	
67	SST-067	7-Jan-16	Test Case Passed	P							P	1	
68	SST-068	7-Jan-16	Test Case Failed	F							F	1	
69	SST-069	7-Jan-16	Test Case Passed	P							P	1	
70	SST-070	7-Jan-16	Test Case Failed	F							F	1	
71	SST-071											0	
72	SST-072											0	
73	SST-073											0	
74	SST-074											0	
75	SST-075											0	

Following are the Base Metrics calculated using the sample data given in Test Execution Matrix above

Base Metrics	
Metric	Value
Total # of TCs	150
No of Test Cases Executed	68
No of Test Cases Passed	47
No of Test Cases Failed	20
No of Test Cases in Observation	1
No of Test Cases Blocked	2
No of Test Cases Not Executed	82
No of Cases Re-executed	1
Total Executions	70
Total Passes	47
Total Failures	22
1st Run Failures	21

Following are the metrics calculated in percentage (%) based on above Base Metrics

Calculated Metrics	
Metric	Value
% Complete	23.3%
% Test Coverage	32.0%
% TCs Passed	72.9%
% TCs Blocked	1.3%
% 1st Run Failures	27.1%
% Failures	28.0%
% Defects Corrected	14.3%
% Rework	100.0%

Following are the actual results manually taken based on Test Execution Status Matrix above.

Actual Results		
Days	Total Passed	Total Failures
1	1	1
2	4	3
3	21	3
4	29	9
5	34	14
6	39	19
7	46	22
8		
9		
10		

S-Curve:

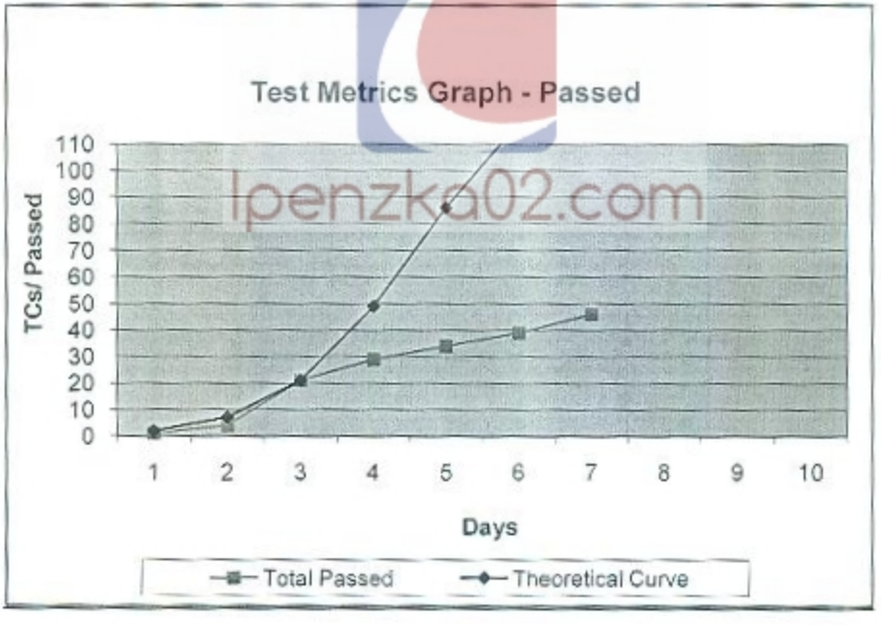
S-Curve is a tool that used in project management to monitor growth, progress and performance of projects undergoing. It is a form of mathematical theory that is represented by a curvature (graph) and illustrates the side-by-side comparisons of the actual results and expected results.

I have used this tool on above Test Execution Matrix and calculated S-Curve values for estimated (theoretical) or expected pass tests.

Following is the S-Curve calculation for estimated (theoretical) or expected passed tests

S-Curve Calculations for Passed Tests		
No of Days	Number of Test Cases	
10	150	
Days	Theoretical Curve	
1	1.10%	2
2	4.70%	7
3	14.14%	21
4	32.82%	49
5	57.61%	86
6	78.40%	118
7	90.41%	136
8	96.00%	144
9	98.36%	148
10	99.33%	149

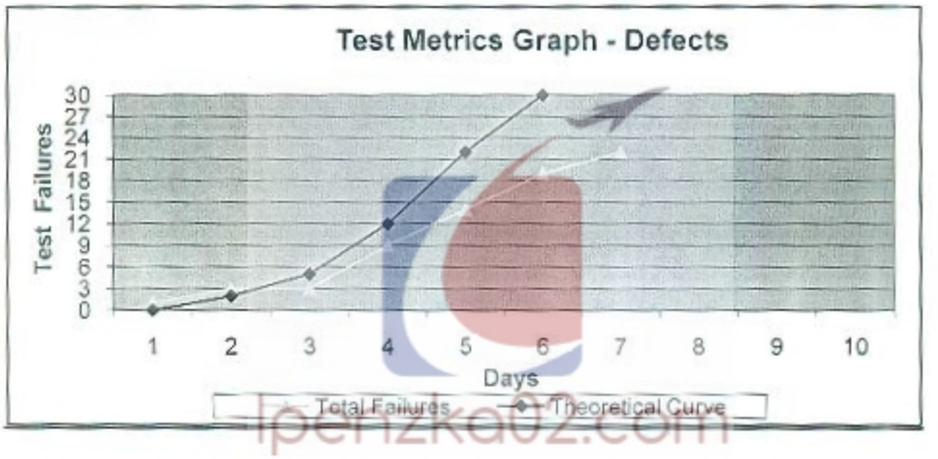
Below S-Curve graph shows the actual passed tests (Total Passed Curve) taken from **Actual Result** table above versus estimated or theoretical passed tests (Theoretical curve)



Following is the S-Curve calculation for estimated or expected Defects

S-Curve Calculations - Defects		
Days	Total Number of Defects	
10	38	
	Theoretical Curve	
1	1.10%	0
2	4.70%	2
3	14.14%	5
4	32.82%	12
5	57.61%	22
6	78.40%	30
7	90.41%	34
8	96.00%	36
9	98.36%	37
10	99.33%	38

Below S-Curve graph shows the actual failure of tests (Total Failure) taken from **Actual Result** table above versus estimated or theoretical failed tests (Theoretical curve)



Here the S-Cure graph helps to determine the progress of testing based on the status of passed and failed tests against the estimated values.

Element Three

A systematic, theory-based formulation of engineering fundamentals required in the engineering discipline

Context

Engineering fundamentals provide the knowledge base for engineering specialisations and represent a systematic formulation of engineering concepts and principles based on mathematical and natural sciences to support applications.

The core areas of engineering fundamentals knowledge include fluid mechanics, statics and dynamics, electric circuits, solid mechanics, thermodynamics, heat transfer, mass transfer, and properties of materials.

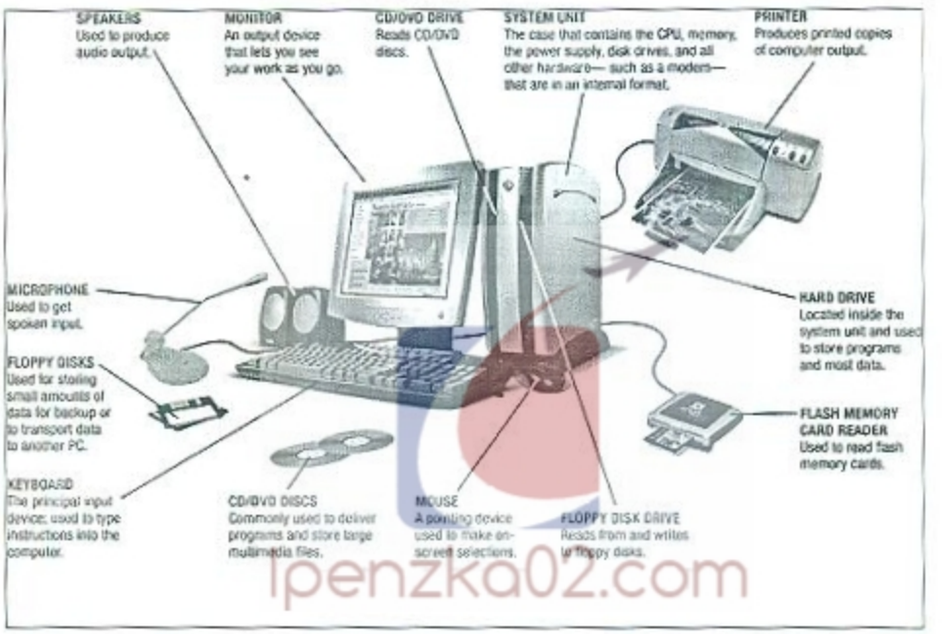
Washington Accord graduates are expected to be able to apply this knowledge of engineering fundamentals to solve complex engineering problems.

Performance Indicators

- Ability to define key factual information in core areas of fundamental engineering knowledge relevant to your engineering discipline.
- Evidence of sufficient depth of knowledge of engineering fundamentals to demonstrate an ability to think rationally and independently within and outside a chosen field of specialisation
- Evidence of sufficient breadth of knowledge of engineering concepts and principles to allow subsequent professional development across a broad spectrum of engineering
- Ability to apply knowledge of engineering fundamentals to solve complex engineering problems relevant to your discipline

	Provide annotations to your supplementary evidence (document and page number)
Summarise your knowledge of the core engineering fundamentals (as listed above) and how they have been developed through formal study, on-job learning and/or continuing professional development. Note: please cross-reference to your academic transcript(s) and continuing professional development records, as appropriate. I gained fundamental computer engineering knowledge during my bachelor programmes; where I studied various core subjects of computer engineering that developed foundation of engineering knowledge and understanding with sufficient depth. These courses had helped me to develop ability to think logically and given me confidence to apply those knowledge during my course work assignment and in on job learning experience later on. These subjects are concerned with concepts, theory, principals, method, skills and application of computing that forms the foundation for the development of software and the discipline of software engineering. **1) Introduction to Computers:** In today's world, computers are used for almost every task that we can imagine. Routine activities such as calculations, documentations, academic work, paying bills, buying groceries, communicating with a friend, all can be done with a computer. Therefore it is important not only to know how to use a computer, but it is also important to understand the components of a computer and what they do and how they work. In this subject, I started to learn from basics of computers. I studied about the main components of computer hardware and gained knowledge about how computer take inputs, process and provide output. I also studied how computer process data using processor (that	BS(Computer Engineering) Transcript Course, CE101

is also called central processing unit-CPU) taking instructions from ROM and RAM, and use its subcomponents called Arthematic Logic Unit, Memory and Control Unit. I also studied about the types of input data that computer received for example instruction or data and types of output for example video, audio, text, graphics. I also studied about the output devices, storage devices (Hard Drive and CD Rom Storage) and the mechanism to calculate the memory by Bit, Byte, KB, MB, GB and TB. In this subject, I also studied about the types of software's and their importance with examples like system software's with their details, application software's, word processors software's, database software's, desktop publishing software's and other types of software's with their sub types. I also studied about number of other applications available. There are software programs that can be used by musicians to produce musical scores and play them on a synthesizer, programs that assist an architect in designing a building, programs that produce the special effects that you see in movies, and programs that allow e-mail or electronic mail. Every line of work you can think of has had applications developed, which can ease or enhance its effectiveness and consistency. I also studied about the types of computers like personal computer, desktop computers, notebook computers, PDA (Personal Digital Assistance), Mainframe computers. This subject gave me fundamentals knowledge of computer communication and networks and internet and it's working. I also studied the types of various users of computers.



In the current world, it is almost impossible to imagine that someone can live without computers. Computers have become an electronic device of almost every day use for individuals of every age. They are essential in almost all the business dealings that are made nowadays. The most that any industry has gained from the discovery of the computers is the business industry because of its nature. Computers have gained significance as they have improved the efficiency and productivity of work done. Large amounts of information in industrial and business sectors as well as in the personal lives are stored on computers. I have my personal computer, laptop and tablet device that I use for my personal and professional work and as a computer engineer, I know everything about it e.g. Memory, Hard Drive, Processor and other portable devices I use. I am also able to setup and configure and troubleshoot related problems and resolve.

- As part of Continuing Education, I taken training course on 'Computer Hardware and Troubleshooting' (2 Month) from 13th September, 2009 to 14th November, 2009 from Rahimabad Ismaili Students Association.

Certificate of Participation in Workshop from RISA

This workshop had given me hands on experience of computer hardware troubleshooting techniques. The knowledge gained has added further value to my overall practical

experience to work with computer hardware and latest advancement in computer hardware technologies.

2) Programming Language Fundamental Areas

BS(Computer Engineering) Transcript

Courses, CE101 & CE103 & CE304 & CE307

In this knowledge Area I studied four subjects

a) Computer Programming and Problem Solving (I & II)

b) Object oriented Programming

c) Basic Compiler Design

Computer Programming and Problem Solving (I & II):

In computer programming, it is learned how to convert problem solutions into instructions for the computer, while doing programming how instructions are prepared in computer program and runs those instructions on the computer, tests the program to see if it is working properly, and makes corrections to the program.

I developed foundation of computer programming by studying this subject and developed logic building ability while creating simple to complex programs as part of my course work assignments and related projects.

Using the language of C and C++ I learned to work with programming fundamentals like (operators and expressions, data types, control flow, function and program structure, pointers and arrays, data structure, memory management, file structure and manipulating files, derived Classes, abstract Classes, design of class hierarchies, class hierarchies and Abstract Classes), and programming syntax and develop various small-scale programs as part of course work.

While studying advance course in computer programming, I also learned various other programming languages to explore other tools and technologies to develop various software applications e.g. At that time I learned to work with C Sharp language and developed Library Management System and School Management System as part of the semester projects. These projects gave me confidence and helped me to work with more complex programming languages and develop relevant skills.

Based on these subjects I also developed skills to analyze software programmes that helped me to work as software quality engineer during my professional career.

Object Oriented Programming (OOP):

Object oriented programming (OOP) is a programming paradigm based on the concept of 'objects, which may contain data, in the form of fields, often known as attributes, and code, in the form of procedures, often known as methods. A feature of objects is that an object's procedures can access and often modify the data fields of the object with which they are associated

In this subject, I learned the foundation of Object Oriented Programming Principals as part of Software engineering. In this subject, I learned to work with Encapsulation, Abstraction, Inheritance and Polymorphism in programming language. During academic study, I worked in C sharp language to implement the principals of Object Oriented Programming and develop various simple and complex programs as part of course work. During professional career, I learned to develop and analyze the enterprise application programmes using the object oriented programming principals that helped me to understand the programme structure and relations between objects among classes. I worked in different roles as Software Quality Engineer in various organizations where I used the skills of OOP to review the system design and programming code and identified various critical and major software bugs in programming code and work with software architect and software developer to fix them in order to stable the product features and system.

Basic Compiler Designing:

Compiler design principles provide an in-depth view of translation and optimization process. Compiler design covers basic translation mechanism and error detection and recovery. It includes lexical, syntax, and semantic analysis as front end, and code generation and optimization as back-end.
In basic Compiler designing, I studied compiler and its application, interpreters, lexical analysis, syntax Analysis, Scope and symbol tables, interpretation, type checking intermediate code generation, Machine code generation, Register allocation and Function call.

3) Data Structure and Algorithm

In this subject, I developed fundamental knowledge and skills related to data structure and algorithms in which I studied basic data structure, algorithm designing and algorithm analysis. I studied about fundamental concepts of data structure like Array Data Structure and its operations (Traverse, Insertion, Deletion, Search, and Update). I also studied about link list (Double Link list, Circular Link List) and its operations. I studied Searching technique (Linear, binary, interpolation, hash table). I studied about various sorting techniques in algorithm like bubble sort, insertion sort, selection sort, merge sort, shell sort and quick sort. I also studied tree data structure in which I learned tree traversal, binary search tree, AVL tree, spanning tree and heap.

BS(Computer Engineering) Transcript

Course, CE106

I studied about the characteristics of algorithm. I which I studied that algorithm should always be unambiguous, it must have input and output, and it should be finiteness and should have feasibility and independent. I also studied about Asymptotic analysis, Greedy Algorithm, Divide and Conquer and Dynamic Programming. I also learn to write algorithm and came to know that there are various solutions to a problem but it should always be implemented after careful analysis (it may be priori analysis and posterior analysis) and complexities.

As Software Quality Assurance Professional, I also developed skills of algorithm analysis that contributed significant value in my analytical skills during my professional career. This subject has helped me to understand and analyzed the implementation of data structure in various enterprise applications during my professional work. While working in Mobile complete, Inc, I explored the code of enterprise application and also analyzed and proposed algorithms for different features.

Example 3-1: Example of Asymptotic Analysis:

Compare the asymptotic analysis for the following two code fragments that I have written in C language

```
sum1 = 0,
for (a=1, a<=n, a++) // 1st double loop
for (b=1, b<=n, j++) // execute n times
sum1++,

sum2 = 0,
for (a=1, a<=n, i++) // 2nd double loop
  for (b=1, b<=a, j++) // do a times
sum2++,
```

In above example, the first double loop having the inner for loop always executes n times because the outer loop executes n times, it should be clear that the statement sum1++ is executed precisely (n power 2) times. The second loop is analogous to the previous example. This is approximately (1/2) npower2. In both double loops, the second requires half time to execute of the first.

4) Software Engineering

Software engineering is the application of engineering to the design, development, implementation, testing and maintenance of software in a systematic method.

BS(Computer Engineering) Transcript

Course. CE202

In this subject I studied about the Software Engineering Applications and Characteristics, Software life cycle stages and various case studies, Software Engineering as a Layered Technology, Software Process Models, Evolutionary Software Process Models, Component Based Development, Project Management Concepts, Project Metrics, Decomposition Techniques, Project Scheduling & Tracking Analysis Modelling, Software Quality Assurance, Software Testing Strategies.

5) Computer Architecture and Organization

In this knowledge Area I studied two subjects

BS(Computer Engineering) Transcript

Course. CE203 & CE309

a) Computer Architecture and Organization

b) Advance Computer Architecture and Organization.

Computer Architecture and Organization:

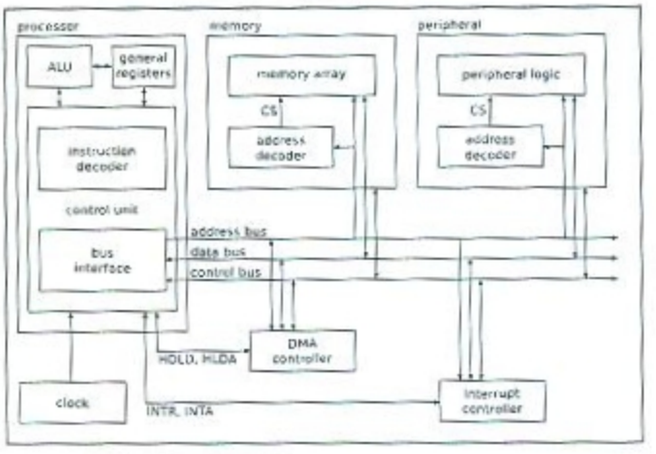
Computer Architecture and Organization is the study of internal working, structuring and implementation of a computer system.

In these subjects, I studied internal working, structure and implementation of a computer system. By studying the subject I was able to understand that design of the computer must contribute to maximize performance while keeping power consumption in control, low cost relative to amount and it should be reliable.

In this subject I studied internal structure of CPU, bus architecture of computers, performance comparison among computer architectures, CPU design, cache memories and its different designs, virtual memory concept and its working, address mapping using pages, pipelining, super scaling threading, instruction level (ILP), parallel processing, branch prediction, pre-fetching and multithreading. Along with this subject, I learned assembly language programming, this is the first machine language that I studied that allowed me to learn various ways to interact with computer hardware and machine code instructions using programs. I learned to write basic Pentium arithmetic and logic instructions, instructions of data movement in flag registers and its effects on the flag, I learned Pentium addressing modes, stack operations and procedure writing. I created various programs of file processing, I/O instructions and work with various BIOS software interrupts.

As a computer engineer, this subject has helped me to understand complete architecture of computer; and it has also added further value in my profession by allowing me to interact with computer hardware and its core components for trouble shooting, learning advance technologies without hesitation.

Diagram of Basic Computer Architecture



6) Communication and Networks:

In this knowledge area I studied three subjects

a) Data Communication

b) Computer Communication and Network,

c) Computer Security

BS(Computer Engineering) Transcript

Course, CE312 & CE402 & CE408

In **Data Communication**, I studied and gained knowledge about Overview of OSI and TCP/IP models, The internet and modern network architecture, Physical layer, Circuit switching and packet switching networks, LAN topologies, Data link layer issues which includes Ethernet standards, collision detection (ALOHA, CSMA/CD), MAC addressing, NAT/PAT, Error and flow control. I also studied HDLC and Point-to-point protocol, Multiplexing and multiple access techniques (FDM, TDM, CDMA, OFDM), The internet protocol (IP), IP sub-netting and addressing, Routing protocols (RIP, IGRP, STP, OSPF), QoS and multicast addressing protocols, Comparison between IPv4 and Ipv6.

In **Computer Communication and Network**, I studied and gained knowledge about Introduction of Computer Networks and Services, Network Design Principles, OSI and TCP/IP Reference Models, Network Topologies, the Physical Layer and Data Communication Fundamentals, Transmission Medias, Data Encoding, Data Communication Interfaces, Data Link Layer and its Protocols, Multiplexing, FDM and TDM, Medium Access Control and Various Multiple Access Methods, Ethernet and Token Ring Systems, Wide Area Networks, Network Layer and Routing, Hub, Bridges and Switches, Internetworking, IP Protocol, IP Addressing, Transport Layer, Services provided by Transport Layer, TCP & UDP, Congestion Control & Quality of Service, Application Layer, Domain Name System, Worldwide Web, Overview of Network Security.

- As part of Continuing Education, I taken certification training (2 Month) of Interconnecting Cisco Network devices to gain further knowledge and practical experience using Cisco labs in July,2003 to August,2003 from Computer Training and Testing Centre (CTTC)

Course completion Certificate- Interconnecting Cisco Network Devices - CTTC

- As part of Continuing Education, I taken certification training (2 Month) of Building Scalable Cisco internetworks to gain further knowledge and practical experience using Cisco labs in August, 2003 to September,2003 from Computer Training and Testing Centre (CTTC)

Course completion Certificate- Building Scalable Cisco Networks- CTTC