

QUALIFICATION FILE

Annexure – I

Course Content for PG-Diploma in System Software Development (PG-DSSD)

Course Modules:

Sr. No	Module	Hours
1	Advanced C/C++ programming	140
2	Computer Networks and Security	80
3	Software Engineering	20
4	Linux System Programming and Driver Development	200
5	Windows System Programming	80
6	Cloud Computing	80
7	Mobile Computing and Android Programming	80
8	Project and Seminar	120
9	Aptitude and Business Communication	100
Total		900

Detailed Contents:

Advanced C/C++ programming (140 hrs)

C Programming

Introduction, Data Types, Operators, Control Statements, Functions, Data Input & Output, Arrays, Pointers, Strings, Structures & Unions, Bit field operators, preprocessors, C and Assembly, Files, I/O, Code Optimization, Secure Coding Practices in C programming

Data Structures

Introduction to Data Structures, ADT, Complexity of Algorithms, Linked Lists, Stacks, Queues, Sorting & Searching Algorithms, Trees & Graphs

C++ Programming

OOPS Concepts, classes, friend functions, Static data members & member functions, Operator and Function Overloading, Inheritance, C++ Files and I/O Support, Virtual Functions and Polymorphism, Templates, Exception Handling, RTTI and STL, Secure

coding practices in C++

Day-wise Syllabus for Embedded C Programming

Session Plan (Each session is of 2 hrs.)

C Programming:

Session	Topic
1	Introduction
2	Data types, Operators
3	Control Statements
4	Functions
5	Data Input & output
6	Arrays
7	Multi-Dimensional Arrays
8	Pointers
9	Pointers and arrays
10	Pointers with functions
11	Strings & Pointers with strings
12	Structures and Unions, Bit field operators
13	Preprocessors, C and Assembly
14	Files and I/O, Code optimization
15	Secure Coding Practices in C

C++ Programming

Session	Topic
1	OOPs concepts, Salient features of object oriented programming
2	Classes, Objects, data members and member functions
3	Friend functions, function overloading static data members and member functions
4	Constructors & Destructor
5	Operator Overloading
6	Inheritance, runtime polymorphism and virtual functions
7	Templates and STL
8	Exception handling, Namespaces and RTTI
9	Console and File I/O
10	Secure coding practices in C++

Data Structures:

Session	Topic
1	Introduction to data structures, complexity of algorithms. Introduction to Stacks. Applications and Implementation
2	Queues, circular queues
3	Linked Lists
4	Doubly Linked Lists
5	Searching Algorithms - Sequential, Binary and Hashing Sorting Algorithms - Selection Sort

6	Sorting Algorithms - Bubble, Insertion, Quick and Merge Sort
7	Trees - Introduction, Binary Search Tree
8	Binary Search Tree
9	AVL trees, Heap sort
10	Introduction to Graphs and applications

Text Books:

1. "C++ Complete Reference", Herbert Schildt, Tata McGraw-Hill
2. "Data Structures and Algorithm Analysis: in C++, Third Edition", Mark Allen Weiss, Addison-Wesley

Reference Books:

1. "The C++ Programming Language", Bjarne Stroustrup, Addison-Wesley
2. Data structures & Algorithms by R.S. Salaria
3. C Programming Language, Schaum Series
4. C Programming Language by Kernighan and Ritchie
5. Algorithms and Data Structures by Niklaus Wirth
6. Intel manual for X86 architecture
7. Introduction to Algorithms by Thomas H. Cormen, Charles E. Leiserson, and Ronald L. Rivest, McGraw-Hill, 1991 (Third Edition)
8. An Introduction to Programming and Problem Solving with Pascal, by Schneider, Weingart and Perlman
9. Structured programming by Ole-Johan Dahl, Edsger Wybe Dijkstra, Charles Antony Richard Hoare
10. Structure and Interpretation of Computer Programs, 2nd edition by Harold Abelson and with Gerald Jay Sussman Julie Sussman

Computer Networks and Security (80 hrs)

Introduction, OSI Layers, TCP/IP protocol suite, underlying technologies, network layer, transport layer & protocols, application layer protocols, Network Packet capturing, Security Engineering Overview, Goals of Security, Types of Cryptographic algorithms, Digital Signatures and certificates, viruses & Intruders, Firewall & Intrusion Detection System, Security Protocols, Application Security, Coding Threats and Malware Analysis.

Session Plan (Each session is of 2 hrs.)

Session	Theory Topic
1	Introduction, OSI Layers, TCP/IP protocol suite
2	Underlying technologies (data link layer)
3	Introduction to Network Layer
4	IPv4 addressing, Delivery and forwarding of IP packets
5	Introduction to Transport Layer
6	Transport Control Protocol and User Datagram Protocol
7	Application Layer - DNS, FTP
8	Application Layer -SMTP, WWW and HTTP and others
9	Network Packet Capturing and Analysis (PCAP programming)
10	Introduction to Security Engineering, Goals of Security

11	Types of cryptographic algorithms
12	Hash functions, Digital certificates and Signatures
13	Viruses and Intruders, Best practices in information security and social issues
14	Firewall and Intrusion detection system
15	Coding threats (buffer overflow, format string vulnerabilities, Integer overflow etc)
16	Application security - Web Proxy Editor, Cookies, Session Management, Forceful browsing, Cross site scripting, SQL Injection, and CAPTCHA
17	Security protocols - SSL, S/MIME, PGP
18	Application security using openSSL
19	Security protocols - Kerberos, overview on IPsec
20	Malware Analysis

Text Books:

1. "TCP/IP Protocol suite", Behrouz Forouzan, 4th edition, Tata McGraw-Hill
2. Stallings, William. Network Security Essentials, Prentice-Hall, 2003

Reference Books:

1. "Computer Networks Fourth Edition", Andrew S. Tannenbaum, Prentice Hall
2. "TCP/IP illustrated, Volume 1: "The protocols", Richard Stevens, Addison-Wesley.
3. Distributed Computing: Principles and Applications by M.L.Liu
4. Internetworking - Vol I by William Comer

Software Engineering (20 hrs)

Overview of Software Engineering: Introduction to Software and Software Engineering, Criteria for a Successful Project, Software Quality Assurance

Software Development Methodologies: Development Life Cycle Models (Waterfall, prototyping, evolutionary development, spiral, synchronize & stabilize, comparing different SDLC models and comparing plan-driven & Agile methodologies), Project Execution Models (On-site, Off-shore & On-site cum off-shore), Software Requirements Engineering (Requirements Analysis, Requirements specifications, Requirements validation, Structure of SRS and characteristics of a good SRS document), Software Design (Design Principles, Structured Vs Object Oriented, design issues, design reviews & walkthroughs, structure of design document), Software Implementation (implementation methodologies – top-down Vs bottom-up, objectives of code writing, coding guidelines & standards, Defensive programming and code verification), Software Testing (Validation & Verification, Levels of Testing – Unit, Integration, System & Acceptance, Functional Testing Vs Structural Testing, Mutation Testing, Regression Testing, Types of Testing - Black box. White box, Gorilla, Beta, Performance, Stress and Acceptance, Acceptance Test Plan, Test Cases, Criteria for completion of Testing, classification of defects, software reliability – MTBF, MTTR & MTTF), Software Release and Configuration Management (Packaging and Releasing the software product, Installation & Commissioning, user training, maintenance, version and release control of source code, software configuration management activities)

Software Project Management: Project Feasibility Studies (Market Analysis / Research, feasibility studies – technical & economic, Risk Analysis), Project proposal (format – technical & commercial aspects), Software Project Estimation (Process, Inputs for project estimation, effort estimation methodologies, Constructive Cost Model (COCOMO)), Software Project Planning & Monitoring (Execution model, review planning, process auditing, risk management planning, project monitoring), Learning from History: Postmortem Analysis (Criteria for completion of projects, Five angles of postmortem analysis – process, people, technology, management & money, postmortem analysis methodology, incorporating analysis results in future efforts)

Software Quality Management: ISO 9000 Quality Management System, Capability Maturity Model for Software, People Capability Maturity Model, Capability Maturity Model Integration for Software Engineering, Personal Software Process

Reference Books:

1. An Integrated Approach to Software Engineering, Third Edition. Author: Pankaj Jalote
2. *Software Engineering: A Practitioner's Approach*, Roger S Pressman, 6th Edition, McGraw-Hill publications
3. *Software Engineering by Sommerville*

Linux System Programming and Driver Development (200 hrs)

Syllabus

OS Goals, Services, Components, OS Architecture, Monolithic & Micro Kernel, Linux Architecture, File and Console I/O Functions, Processes, Threads and Scheduling, Inter-Process Communication and Synchronization (Shared Memory, Message Queues, Semaphores and Mutex), Socket Programming, Memory Management (X86 specific), File System Management, Kernel Introduction and module programming, Introduction to Device Drivers, Char Drivers, Synchronization, Debugging techniques, Hardware Management and Interrupt Handling (Top halves and Bottom halves), Timers, taskqueues, tasklets, Kernel Data types, USB drivers, Overview on Block Drivers and Network Drivers

Session Plan (Each session is of 2 hrs.)

Linux System Programming

Day	Session - I	Session - II
1	Introduction to OS & System calls	Linux OS Installation
2	Shell Programming	Shell Programming
3	Python Scripting	Python Scripting
4	File System Management	GNU Tool Chain
5	Process Management	Process Management
6	Files & Environmental Variables	Process Scheduling
7	Synchronization	Synchronization
8	Threads	Threads & Synchronization
9	Deadlocks	Pipes, Named Pipes

10	IPC Mechanisms	IPC Mechanisms
11	Socket Programming	Socket Programming
12	Socket Programming	Socket Programming
13	Memory Management	Memory Management
14	Memory Management (x86 specific)	Signals
15	I/O systems	I/O systems

Linux Driver Development

Day	Session - I	Session - II
1	Introduction	Module Programming
2	Module Programming	Module Programming
3	Character drivers	Character drivers - Read & write
4	Advanced char operations - ioctl	Kernel Linked List
5	Synchronization	Synchronization
6	Parallel Port	Parallel port
7	Interrupt Processing	Kernel Debugging
8	Kernel Compilation & Configuring Kernel for SMP systems	Kernel Timers
9	Memory, Data types and Coding styles	USB drivers
10	Overview on block drivers	Overview on network drivers

Text Books:

1. "Beginning Linux Programming", Neil Matthew, Richard Stones, Alan Cox, Wrox Publications
2. "Linux Device Drivers", Alessandro Rubini, Jonathan Corbet, Third Edition
3. "Linux Kernel Development" by Robert Love
4. "Operating System Concepts", Avi Silberschatz, Peter Baer Galvin, Greg Gagne, John Wiley & Sons, Inc.

Reference Books:

1. "Tim Burke, Mark A. Parenti, and Al Wojtas, Writing Device Drivers, Tutorial and Reference, Butterworth-Heinemann, July, 1995
2. George Pajari, Writing UNIX Device Drivers (1st edition), Addison-Wesley Professional, November, 1991
3. Mel Gorman, Understanding the Linux Virtual Memory Manager, Prentice Hall, April 29, 2004
4. Linux Internals by Oreilly Publications
5. Distributed Operating Systems: Concepts and Design by P.K.Sinha
6. The Art of Multiprocessor Programming by Maurice Herlihy, Nir Shavit

Windows System Programming (80 hrs)

Syllabus

Windows Programming Overview, Handling Processes and Pipes, Handling Virtual Memory, Unicode and Character Internationalization, Creating and Using Threads, Handling Synchronized Resources, Socket Programming, Using the Registry, Using Named Pipes and Mail Slots, Dynamic Link Libraries, Creating Windows NT Services, kernel mode programming and debugging.

Session Plan (Each session is of 2 hrs.)

Day	Session - I	Session - II
1	Introduction, Windows architecture and Kernel object	Process and Thread (Basics)
2	Thread scheduling, Affinity and priority, Thread Synchronization	Memory Management (stack, Heap, Virtual Memory APIs)
3	IPC Mechanisms (Named Pipe, Mailslot)	DLL (Static, Dynamic)
4	Security Descriptor	Registry, Services and Exception Handling
5	Socket Programming	User Mode Debugging Tools in Windows
6	Introduction to Windows Drivers	Kernel Mode Debugging and Crash Analysis
7	Process and Threads Internal Data Structures)	Memory Management Internals – Paging etc.
8	Read and Write Operations	Device Extensions
9	Device and Driver Stack, IRQL and IRPs	IOCTL
10	Linked Lists in Kernel	Sharing Events between UM and KM

Text Books:

1. "Windows Internals", 6 Edition Parts 1 and 2, David A. Solomon, Mark Russinovich, Alex
2. "Windows System Programming", Johnson M. Hart, Addison Wesley Professional.
(or)
"Windows Via C/C++", Jeffrey Richter, Christophe Nasarre, Microsoft Press.
3. "Windows Driver Model", Walter Oney
(or)
Windows 2000 Device Driver Book, Art Baker
(or)
Writing Windows Device Driver Book, Yashwant Kanetkar

Reference Books:

1. "Programming the Microsoft® Windows® Driver Model, Second Edition", Walter Oney, 0-7356-1803-8, Microsoft Press
2. "Programming Server-Side Applications for Microsoft Windows 2000", Jeffrey Richter, Microsoft Press

3. "Windows System Programming 3rd Edition", Johnson M. Hart, Addison Wesley Professional
4. "Multithreading Applications in Win32: The Complete Guide to Threads", Jim Beveridge, Robert Wiener, Addison-Wesley
5. "Microsoft Windows Internals, Fourth Edition", Mark E. Russinovich, David A. Solomon, Microsoft Press
6. "Programming Applications for Microsoft Windows", Jeffrey Richter, Microsoft Press
7. "Windows Wdm And Nt5 Device Drivers: Master The New Driver Model For: Windows 98 / Windows 2000", by Chris Cant, Cmp Books publishers.
8. "Developing Drivers With The Windows Driver Foundation" by Penny Orwick, Guy Smith. Dreamtech Press publishers.

Cloud Computing (80 hrs)

Syllabus

Introduction to parallel Programming, OpenMP, MPI, Importance and fundamentals of Virtualization, Virtualization Tools and Technologies, Containers, LXC, Docker, Introduction to Cloud computing, cloud Architecture, Open Stack, Open stack Developer's API.

Session Plan (Each session is of 2 hrs.)

Day	Session - I	Session - II
1	Introduction to parallel Programming	OpenMP
2	OpenMP, MPI	MPI
3	Virtualization - Introduction	Virtualization Tools and Technologies
4	Virtualization Tools and Technologies	Virtualization Tools and Technologies
5	Containers, LXC, Docker	Containers, LXC, Docker
6	Introduction to Cloud Computing	Architecture
7	Introduction to Open Stack	Installation and configuring Open Stack
8	Open stack Developer's API	Open stack Developer's API
9	Cloud application development	Cloud application development
10	Cloud application development	Cloud application development

Reference Books:

1. Shameem Akhter, Jason Roberts, " Multi-Core Programming " , Intel Press
2. Nathan Brookwood, "The Role of Intelligent Design in the Evolution of Multi-Core Processors" , Insight 64
3. Dr.Dobbs Journal, " Multi-Threaded Technology & Multi-core Processors", May 2006
4. MPI—The Complete Reference Volume 1, The MPI Core Second Edition by Marc Snir, Steve Otto, Steven Huss-Lederman, David Walker, Jack Dongarra
5. "OpenCL Programming Guide"- Aaftab Munshi; Benedict R. Gaster; Timothy G. Mattson; James Fung; Dan Ginsburg Addison - Wesley Professional
6. "OpenCL Programming Book"- RyojiTsuchiyama, Takashi Nakamura, TakuroIizuka, Akihiro Asahara, Satoshi Miki - Fixstars Corporation
7. "Heterogeneous Computing with OpenCL" - Benedict Gaster, Lee Howes , David R. Kaeli , PerhaadMistry , Dana Schaa - Morgan Kaufmann Publishers

8. An Introduction to Parallel Programming by Peter Pacheco
9. "Parallel Programming in C with MPI and OpenMP" by Michael Quinn

Mobile Computing and Android Programming (80 hrs)

Syllabus

Introduction to Java, Creating Classes, Objects, Inheritance, Interfaces & Packages, Exceptions, Multithreaded Programming & Synchronized methods, Input & output stream classes, Overview of Java Native Interface (JNI) and JNI components.

Introduction to Mobile Computing, Overview of Android Platform& Android Building Blocks, Overview of Android Application Components (Activity, Intents, Broadcast Receiver, Content Providers, Service), Android GUI Development and Event Handling, Android Permission Model and Application Sandboxing Cross-Compiling System Programs for Android Platforms, Overview of Android Native Development Kit (NDK) Advantages and Disadvantages of using NDK, Java Native Interface (JNI) Using NDK.

Session Plan (Each session is of 2 hrs.)

Day	Session - I	Session - II
1	Introduction to Java, Creating Classes, Objects, Inheritance	Introduction to Java, Creating Classes, Objects, Inheritance
2	Interfaces & Packages, Exceptions	Interfaces & Packages, Exceptions
3	Multithreaded Programming & Synchronized methods	Multithreaded Programming & Synchronized methods
4	Input & output stream classes	Input & output stream classes
5	Overview of Java Native Interface (JNI) and JNI components	Overview of Java Native Interface (JNI) and JNI components
6	Introduction to Mobile Computing, Overview of Android Platform& Android Building Blocks	Overview of Android Application Components (Activity, Intents, Broadcast Receiver, Content Providers, Service)
7	Android GUI Development and Event Handling	Android GUI Development and Event Handling
8	Android GUI Development and Event Handling	Programming on Android Application components
9	Programming on Android Application components	Android Permission Model and Application Sandboxing
10	Overview of Android Native Development Kit (NDK) Advantages and Disadvantages of using NDK	Java Native Interface (JNI) Using NDK

Text Books:

- Professional Android 4 Application Development by Reto Meier Wrox Press

Reference Books:

Java - The Complete Reference, Seventh Edition, Herbert Schildt

Project and Seminar (120 hrs)

- Project work has to be initiated immediately after basic modules, so that they can focus on literature survey, requirements gathering & design in parallel with the remaining modules of the course. Implementation & testing can be done at the end of the course during the 5 weeks period assigned to project.
- The project work is allotted 80 marks and seminar is allotted another 20 marks which a student has to deliver during the conduction of the course.
- For this, students need to choose the topic themselves and give the seminar on the respective dates allocated by the concerned faculty members. The topic chosen by the students should be relevant to the System Software Development. Encourage students select the seminar topics from journals of ACM/ IEEE.
- Project work is distributed in the following phases:
 - Study & Requirements gathering
 - Design
 - Implementation
 - Testing
 - Project report
 - Presentation
- Students need to submit a project report at the conclusion of the project. The students should maintain a logbook, which contains their day-to-day activities during the project phases. The project guide allocated for that project should sign this logbook regularly.
- Enable students' access to important journals of ACM/IEEE like ACM communications, IEEE Micro etc.

Note: In addition to student seminars, special talks related to testing & debugging, Python scripting, Virtualization, Cloud computing etc can be arranged during the course