

# Chhattisgarh Swami Vivekanand Technical University

## Bhilai (C.G.)

### Scheme of Teaching and Examination

#### B.E. V Semester (Computer Science & Engineering)

S. No	Board of Study	Subject Code	Subject Name	Periods per week			Scheme of Exam			Total Marks	Credit L+(T+P)/2
				L	T	P	Theory/Practical				
							ESE	CT	TA		
1	Computer Science & Engg	322551(22)	Microprocessor and Interfaces	4	1	-	80	20	20	120	5
2	Computer Science & Engg	322552(22)	Analysis and Design of Algorithms	3	1	-	80	20	20	120	4
3	Computer Science & Engg	322553(22)	Programming in Java	3	1	-	80	20	20	120	4
4	Computer Science & Engg	322554(22)	Theory of Computation	3	1	-	80	20	20	120	4
5	Computer Science & Engg	322555(22)	Unix & Shell Programming	3	1	-	80	20	20	120	4
6	Computer Science & Engg	322556(22)	Database Management System	3	1	-	80	20	20	120	4
7	Computer Science & Engg	322561(22)	Microprocessor and Interfaces Lab	-	-	3	40		20	60	2
8	Computer Science & Engg	322562(22)	Programming in Java Lab	-	-	3	40		20	60	2
9	Computer Science & Engg	322563(22)	Unix & Shell Programming Lab	-	-	3	40		20	60	2
10	Computer Science & Engg	322564(22)	DBMS Lab	-	-	3	40		20	60	2
11	Humanities	300565(46)	Personality Development	-	-	2			20	20	1
12	Computer Science & Engg	322566(22)	* Practical Training Evaluation & Library	-	-	1			20	20	1
<b>TOTAL</b>				<b>19</b>	<b>6</b>	<b>15</b>	<b>640</b>	<b>120</b>	<b>240</b>	<b>1000</b>	<b>35</b>

**L: Lecture**                      **T: Tutorial**                      **P: Practical**  
**ESE: End Semester Examination**      **CT: Class Test**                      **TA: Teachers Assessment**

*\*To be completed after IV Semester and before the commencement of V Semester*

# Chhattisgarh Swami Vivekanand Technical University, Bhilai

Branch: **Computer Science & Engineering**

Subject: **Microprocessor & Interfaces**

Total Theory Periods: **40**

No. of class Tests to be conducted: **2 (Minimum)**

ESE Duration: **Three Hours**

Semester: **V**

Code: **322551 (22)**

Total Tutorial Periods: **10**

No. of assignments to be submitted: **One per Unit**

Maximum Marks in ESE: **80**

Minimum Marks in ESE: **28**

**COURSE OBJECTIVE:** To introduce the basic concepts of microprocessor & assembly language programming.

## COURSE OUTCOME:

1. The student will be able to analyse, specify, design, write and test assembly language programs of moderate complexity.
2. The student will be able to select an appropriate 'architecture' or program design to apply to a particular situation; e.g. an interrupt-driven I/O handler for a responsive real-time machine. Following on from this, the student will be able to design and build the necessary programs.

**UNIT I: Introduction to Basic Microprocessors:** Historical Background, the Harvard and Princeton architecture, The Microprocessor-Based Personal Computer Systems. The Microprocessor 8085, 8088 basics and comparison (Block & Pin diagram only).

**UNIT II: Microprocessor Architecture 8086:** 8086 basic block diagram, Internal Microprocessor Architecture, Real Mode Memory Addressing, Registers, pin configuration, segmentation.  
**Data Movement Instructions:** MOV, PUSH/POP, Load-Effective Address, String Data Transfers, Miscellaneous Data Transfer Instructions, Segment Override Prefix, Assembler Details. Arithmetic and Logic Instructions: Addition, Subtraction and Comparison, Multiplication and Division, BCD and ASCII Arithmetic, Basic Logic Instructions, Shift and Rotate, String Comparisons. Program Control Instructions: The Jump Group, Controlling the Flow of the Program, Procedures, and Introduction to Interrupts, Machine Control and Miscellaneous Instructions. Assembler directives, assembler instructions, Assembly Language Programming.

**UNIT III: Assembly Language programming with C/C++, Interrupt and Timing diagrams:** Using Assembly Language with C/C++ for linking C/C++ into assembly language, Basic Programs - Use of BIOS and DOS Interrupts in assembly & C/C++, Interrupts of 8086 microprocessors, Timing diagram of 8086 microprocessor.

**UNIT IV: Memory and I/O Interfacing:** Minimum and Maximum mode configuration of 8086, Memory Interface with 8086 microprocessor, Address Decoding. Basic I/O Interface: Introduction to I/O Interface, I/O Port Address Decoding. I/O Interface using peripheral devices: The Programmable Peripheral Interface 8255, Programmable Interval Timer 8254. **Direct Memory Access:** Basic DMA Operation and Definition.

**UNIT V: Advanced Microprocessors:** 80386- Features, block diagram, data types, supported registers, memory system, real mode and protected mode operation, descriptors, cache register, control register, paging mechanism, virtual mode, and protection mechanism for operating system..  
**Comparative Study of Modern Microprocessor (Web based Reference for study):** Pentium Pro (Pentium II, Pentium III, Pentium IV), Core i3,i5,i7 and Atom processors.

## Text Book:

1. Barry B Brey: The Intel Microprocessors, 8th Edition, Pearson Education, 2009. (Listed topics only from the Chapters 1 to 13)
2. Ramesh S. Gaonkar : Microprocessor Architecture, programming and Application with 8085, 4<sup>th</sup> Edition, Wiley,2012

## Reference Books:

1. Douglas V. Hall: Microprocessors and Interfacing, Revised 2<sup>nd</sup> Edition, TMH, 2006.
2. James L. Antonakos: The Intel Microprocessor Family: Hardware and Software Principles and Applications, Cengage Learning, 2007.
3. Nilesh B. Bahadure: Microprocessors: The 8086/8088, 80186/80286, 80386/80486 and the Pentium family, 2<sup>nd</sup> edition (2014), Prentice Hall of India (PHI).
4. K. Udaya Kumar & B.S. Uma Shankar: Advanced Microprocessors & IBM-PC Assembly Language Programming, TMH 2003.
5. Microprocessor: Theory and Applications- Intel and Motorola, Rafiquzzaman, PHI.

# Chhattisgarh Swami Vivekanand Technical University, Bhilai

Branch:	<b>Computer Science &amp; Engineering</b>	Semester:	<b>V</b>
Subject:	<b>Analysis &amp; Design of Algorithms</b>	Code:	<b>322552 (22)</b>
Total Theory Periods:	<b>40</b>	Total Tutorial Periods:	<b>10</b>
No. of class Tests to be conducted:	<b>2 (Minimum)</b>	No. of assignments to be submitted:	<b>One per Unit</b>
ESE Duration:	<b>Three Hours</b>	Maximum Marks in ESE:	<b>80</b>
		Minimum Marks in ESE:	<b>28</b>

## COURSE OBJECTIVE:

- This course aims to introduce the classic algorithms in various domains, and techniques for designing efficient algorithms.
- Introducing students to the general tools and techniques for analysing computer algorithms.
- Equip the students with mathematical preliminaries required to analyse and design computer algorithms.

## COURSE OUTCOME:

After successful completion of this course, students should be able to:

- Prove the correctness and analyze the running time of the basic algorithms for those classic problems in various domains.
- Apply the algorithms and design techniques to solve problems.
- Analyze the complexities of various problems in different domains.

- UNIT I**     **Algorithms:** Definitions and notations: standard notations - asymptotic notations – worst case, best case and average case analysis; big oh, small oh, omega and theta notations; Recursive algorithms, analysis of non-recursive and recursive algorithms, solving recurrence equations, analysing control structures. Analysis of Sorting and Searching: Heap, shell, radix, insertion, selection and bubble sort; sequential, binary and Fibonacci search.
- UNIT II**     **Divide and Conquer Method:** General Method – binary search – finding maximum and minimum – merge sort and quick sort – Strassen’s Matrix multiplication. Greedy Method: General method – knapsack problem – minimum spanning tree algorithms – single source shortest path algorithm – scheduling, optimal storage on tapes, optimal merge patterns.
- UNIT III**     **Dynamic Programming:** General method – multi-stage graphs – all pair shortest path algorithm – 0/1 Knapsack and Travelling salesman problem – chained matrix multiplication – approaches using recursion – memory functions. Basic Search and Traversal technique: Techniques for binary trees and graphs – AND/OR graphs – biconnected components – topological sorting.
- UNIT IV**     **Backtracking:** Back tracking and Recursive back tracking, applications of Back tracking paradigm, The general method – 8-queens problem – sum of subsets – graph coloring – Hamiltonian cycle – Knapsack problem.
- UNIT V**     **Branch and Bound Method:** Least Cost (LC) search – the 15-puzzle problem – control abstractions for LC-Search – Bounding – FIFO Branch-and-Bound - 0/1 Knapsack problem – travelling salesman problem. Introduction to NP-Hard and NP-Completeness, Cook’s theorem.

## TEXTBOOK

1. Ellis Horowitz, Sartaj Sahni, Sanguthevar Rajasekaran, “Fundamentals of Computer Algorithms”, Galgotia Publications Pvt. Ltd., 2008.
2. Cormen, Lelerson, Rivert, “Introduction to Algorithms”, Second Edition, PHI.

## REFERENCES

1. Paneerselvam, “Design and Analysis of Algorithms”, Prentice-Hall of India, 2006
3. Anany Levitin, “Introduction to the Design and Analysis of Algorithms”, Pearson Education, 2005.
4. Gilles Brassard and Paul Bratley, “Fundamentals of Algorithms”, Prentice-Hall of India, 1997. (UNIT I)
5. Thomas H. Corman, Charles E. Leiserson, Ronald and L. Rivest, “Introduction to Algorithms”, Prentice-Hall of India, Second edition, 2003.
6. Richard Johnsonbaugh and Marcus Schaefer, “Algorithms”, Pearson Education, 2004. (UNIT V)

# Chhattisgarh Swami Vivekanand Technical University, Bhilai

Branch:	<b>Computer Science &amp; Engineering</b>	Semester:	<b>V</b>		
Subject:	<b>Programming in Java</b>	Code:	<b>322553 (22)</b>		
Total Theory Periods:	<b>40</b>	Total Tutorial Periods:	<b>10</b>		
No. of class Tests to be conducted:	<b>2 (Minimum)</b>	No. of assignments to be submitted:	<b>One per Unit</b>		
ESE Duration:	<b>Three Hours</b>	Maximum Marks in ESE:	<b>80</b>	Minimum Marks in ESE:	<b>28</b>

## COURSE OBJECTIVE:

- Make them learn about Java programming concepts, graphical user interfaces, basic data structures.

## COURSE OUTCOME:

- Can develop solutions for a range of problems using object-oriented programming.
- Be able to implement, compile, test and run Java programs comprising more than one class, to address a particular software problem.
- Demonstrate the ability to use simple data structures like arrays in a Java program.

- UNIT I**     **Introduction:** Introduction to Java and Java programming Environment. Object Oriented Programming. Fundamental Programming Structure: Data Types, variable, Typecasting Arrays, Operators and their precedence. **Control Flow:** Java's Selection statements (if, switch, iteration, statement, while, do-while, for, Nested loop) Concept of Objects and Classes, Using Exiting Classes building your own classes, constructor overloading, static, final, this keyword
- UNIT II**     **Inheritance:** Using Super to Call Super class constructor, Method overriding, dynamic method Dispatch, Using Abstract Classes, Using final with inheritance. The Object Class. **Packages & Interfaces:** Packages, Access Protection, Importing package, Interface, Implementing Interfaces, variables in Interfaces, Interfaces can be extended. **Exception Handling:** Fundamentals, Types Checked, Unchecked exceptions, Using try & catch, Multiple catch, throw, throws, finally, Java's Built in exceptions, user defined exception.
- UNIT III**     **Multi-Threading:** Java Thread Model, Thread Priorities, Synchronization, Creating a thread, Creating Multiple threads, Using isAlive () and join (), wait () & notify (). **String Handling:** String constructors, String length, Character Extraction, String Comparison, Modifying a string. **Java I/O:** Classes & Interfaces, Stream classes, Byte streams, Character streams, Serialization.
- UNIT IV**     **Applets:** Basics, Architecture, Skeleton, The HTML APPLET Tag, Passing Parameters to Applets, Applet context and show documents (). **Event Handling:** Delegation Event model, Event Classes, Event Listener Interfaces, Adapter classes. **JDBC:** Fundamentals, Type I, Type II, Type III, Type IV drivers. **Networking:** Basics, Socket overview, Networking classes, & interfaces, TCP/IP client sockets, whois, URL format, URL connection, TCP/IP Server Sockets.
- UNIT V**     **AWT:** AWT Classes window fundamentals, component, container, panel, Window, Frame, Canvas, Creating a frame window in an Applet, working with Graphics, Control Fundamentals, Layout managers, Handling Events by Extending AWT components. Core java API package, reflection, Remote method Invocation (RMI) **Swing:** J applet, Icons & Labels, Text fields, Buttons, Combo boxes, Tabbed panes, Scroll panes, Trees, Tables. **Exploring Java-lang:** Simple type wrappers, Runtime memory management, object (using clone () and the cloneable Interface), Thread, Thread Group, Runnable.

## TEXT BOOKS:

1. Introduction to Java Programming: Liang, Pearson Education, 7<sup>th</sup> Edition.
2. Java The complete reference: Herbert Schildt, TMH, 5<sup>th</sup> Edition.

## REFERENCE BOOKS:

1. Balguruswamy, Programming with JAVA, TMH.
2. Programming with Java: Bhave & Patekar, Pearson Education.
3. Big Java: Horstman, Willey India, 2<sup>nd</sup> Edition.
4. Java Programming Advanced Topics: Wigglesworth, Cengage Learning.
5. Java How to Program: H.M. Deitel & Paul J. Deitel, PHI, 8<sup>th</sup> Edition

# Chhattisgarh Swami Vivekanand Technical University, Bhilai

Branch:	<b>Computer Science &amp; Engineering</b>	Semester:	<b>V</b>
Subject:	<b>Theory of Computation</b>	Code:	<b>322554 (22)</b>
Total Theory Periods:	<b>40</b>	Total Tutorial Periods:	<b>10</b>
No. of class Tests to be conducted:	<b>2 (Minimum)</b>	No. of assignments to be submitted:	<b>One per Unit</b>
ESE Duration:	<b>Three Hours</b>	Maximum Marks in ESE:	<b>80</b>
		Minimum Marks in ESE:	<b>28</b>

## COURSE OBJECTIVE:

- Students will learn about a variety of issues in the mathematical development of computer science theory, particularly finite representations for languages and machines.
- Students will gain a more formal understanding of algorithms and procedures.

## COURSE OUTCOME:

At the end of this course students will:

- Be able to construct finite state machines and the equivalent regular expressions.
- Be able to prove the equivalence of languages described by finite state machines and regular expressions.
- Be able to construct pushdown automata and the equivalent context free grammars.
- Be able to prove the equivalence of languages described by pushdown automata and context free grammars.
- Be able to construct Turing machines and Post machines.

**UNIT I THE THEORY OF AUTOMATA :** Introduction to automata theory, Examples of automata machine, Finite automata as a language acceptor and translator. Deterministic finite automata. Non deterministic finite automata, finite automata with output (Mealy Machine. Moore machine). Finite automata with ? moves, Conversion of NFA to DFA by Arden's method, Minimizing number of states of a DFA. Myhill Nerode theorem, Properties and limitation of FSM. Two way finite automata. Application of finite automata.

**UNIT II REGULAR EXPRESSIONS :** Regular expression, Properties of Regular Expression. Finite automata and Regular expressions. Regular Expression to DFA conversion & vice versa. Pumping lemma for regular sets. Application of pumping lemma, Regular sets and Regular grammar. Closure properties of regular sets. Decision algorithm for regular sets and regular grammar.

**UNIT III GRAMMARS:** Definition and types of grammar. Chomsky hierarchy of grammar. Relation between types of grammars. Role and application areas of grammars. Context free grammar. Left most linear & right most derivation trees. Ambiguity in grammar. Simplification of context free grammar. Chomsky normal form. Greibach normal form, properties of context free language. Pumping lemma from context free language. Decision algorithm for context tree language.

**UNIT IV PUSH DOWN AUTOMATA AND TURING MACHINE:** Basic definitions. Deterministic push down automata and non deterministic push down automata. Acceptance of push down automata. Push down automata and context free language. Turing machine model. Representation of Turing Machine Construction of Turing Machine for simple problem's. Universal Turing machine and other modifications. Church's Hypothesis. Post correspondence problem. Halting problem of Turing Machine

**UNIT V COMPUTABILITY:** Introduction and Basic concepts. Recursive function. Partial recursive function. Partial recursive function. Initial functions, computability, A Turing model for computation. Turing computable functions, Construction of Turing machine for computation. Space and time complexity. Recursive enumerable language and sets.

## Text Books :

1. Theory of Computer Science (Automata Language & Computation), K.L.P. Mishra and N. Chandrasekran, PHI.
2. Introduction to Automata theory. Language and Computation, John E. Hopcroft & Jeffery D. Ullman, Narosa Publishing House.

## Reference Books :

1. Finite Automata and Formal Languages: A Simple Approach, A.M. Padma Reddy, Pearson Education, India.
2. Theory of Automata and Formal Language, R.B. Patel & P. Nath, Umesh Publication.
3. An Introduction and finite automata theory, Adesh K. Pandey, TMH.
4. Theory of Computation, AM Natrajan. Tamilarasi, Bilasubramani, New Age International Publishers.

# Chhattisgarh Swami Vivekanand Technical University, Bhilai

Branch:	<b>Computer Science &amp; Engineering</b>	Semester:	<b>V</b>
Subject:	<b>UNIX &amp; Shell Programming</b>	Code:	<b>322555 (22)</b>
Total Theory Periods:	<b>40</b>	Total Tutorial Periods:	<b>10</b>
No. of class Tests to be conducted:	<b>2 (Minimum)</b>	No. of assignments to be submitted:	<b>One per Unit</b>
ESE Duration:	<b>Three Hours</b>	Maximum Marks in ESE:	<b>80</b>
		Minimum Marks in ESE:	<b>28</b>

## COURSE OBJECTIVE:

- To familiarize students with the Linux environment.
- To learn the fundamentals of shell scripting/programming
- To familiarize students with basic Linux shell script programming.

## COURSE OUTCOME:

At the end of this course students will be able to Work confidently in Unix/Linux environment

<b>UNIT I</b>	<b>INTRODUCTION TO UNIX:</b> Architecture of Unix, Features of Unix , Unix Commands – PATH, man, echo, printf, script, passwd, uname, who, date, stty, pwd, cd, mkdir, rmdir, ls, cp, mv, rm, cat, more, wc, lp, od, tar, gzip <b>UNIX UTILITIES:</b> Introduction to unix file system, vi editor, file handling utilities, security by file permissions, process utilities, disk utilities, networking commands, unlink, du, df, mount, umount, find, unmask, ulimit, ps, w, finger, arp, ftp, telnet, rlogin. Text processing utilities and backup utilities , detailed commands to be covered are tail, head, sort, nl, uniq, grep, egrep, fgrep, cut, paste, join, tee, pg, comm, cmp, diff, tr, awk, cpio
<b>UNIT II</b>	<b>INTRODUCTION TO SHELLS:</b> Unix Session, Standard Streams, Redirection, Pipes, Tee Command, Command Execution, -Line Editing, Quotes, Command Substitution, Job Control, Aliases, Variables, Predefined Variables, Options, Shell/Environment Customization. <b>FILTERS:</b> Filters and Pipes, Concatenating files, Display Beginning and End of files, Cut and Paste, Sorting, Translating Characters, Files with Duplicate Lines, Count Characters, Words or Lines, Comparing Files.
<b>UNIT III</b>	<b>GREP:</b> Operation, grep Family, Searching for File Content. <b>SED:</b> Scripts, Operation, Addresses, commands, Applications, grep and sed. <b>AWK:</b> Execution, Fields and Records, Scripts, Operations, Patterns, Actions, Associative Arrays, String Functions, String Functions, Mathematical Functions, User – Defined Functions, Using System commands in awk, Applications, awk and grep, sed and awk.
<b>UNIT IV</b>	<b>INTERACTIVE KORN SHELL:</b> Korn Shell Features, Two Special Files, Variables, Output, Input, Exit Status of a Command, eval Command, Environmental Variables, Options, Startup Scripts, Command History, Command Execution Process. <b>KORN SHELL PROGRAMMING:</b> Basic Script concepts, Expressions, Decisions: Making Selections, Repetition, special Parameters and Variables, changing Positional Parameters, Argument Validation, Debugging Scripts, Script Examples.
<b>UNIT V</b>	<b>INTERACTIVE C SHELL:</b> C shell features, Two Special Files, Variables, Output, Input, Exit Status of a Command, eval Command, Environmental Variables, On-Off Variables, Startup and Shutdown Scripts, Command History, Command Execution Scripts. <b>C SHELL PROGRAMMING:</b> Basic Script concepts, Expressions, Decisions: Making Selections, Repetition, special Parameters and Variables, changing Positional Parameters, Argument Validation, Debugging Scripts, Script Examples. <b>FILE MANAGEMENT:</b> File Structures, System Calls for File Management system, INODES, Structure of Regular File, Directories, Conversions of a Path, name to an INODE, Super Block, INODE Assignment to a New File create, open, close, read, write, lseek, link, symlink unlink, stat, fstat, lstat, chmod, chown, Directory API – opendir, readdir, closedir, mkdir, rmdir, umask.

## Text Books:

1. Unix and shell Programming, Behrouz A. Forouzan, Richard F. Gilberg. Thomson
2. Your Unix the ultimate guide, Sumitabha Das, TMH. 2<sup>nd</sup> Edition.

## References:

1. Unix for programmers and users, 3rd edition, Graham Glass, King Ables, Pearson education.
2. Unix programming environment, Kernighan and Pike, PHI. / Pearson Education
3. The Complete Reference Unix, Rosen, Host, Klee, Farber, Rosinski, Second Edition, TMH.
4. Unix Shell programming, Yashwanth Kanitkar, 1st Edition, BPB Publisher

# Chhattisgarh Swami Vivekanand Technical University, Bhilai

Branch: **Computer Science & Engineering**

Semester: **V**

Subject: **Data Base Management System**

Code: **322556 (22)**

Total Theory Periods: **40**

Total Tutorial Periods: **10**

No. of class Tests to be conducted: **2 (Minimum)**

No. of assignments to be submitted: **One per Unit**

ESE Duration: **Three Hours**

Maximum Marks in ESE: **80** Minimum Marks in ESE: **28**

## COURSE OBJECTIVE:

- To understand the role of a database management system and its users in an organization.
- To understand database concepts, including the structure and operation of the relational data model.
- Can successfully apply logical database design principles, including E-R diagrams and database normalization.
- Construct simple and moderately advanced database queries using Structured Query Language (SQL).
- To understand the concept of transaction, its properties and how to persist the data in complex concurrent users environment.

## COURSE OUTCOME:

- Will be able to describe the basic concepts of RDMBS and relational data model
- Be familiar with the relational database theory & be able to write relational algebra expressions for queries
- Understand DML, DDL and will be able to construct queries using SQL by knowing the importance of data & its requirements in any applications.
- Be familiar with the basic issues of transaction, its processing and concurrency control.
- Be familiar with basic database storage structures and access techniques: file and page organizations, indexing methods including B-tree, and hashing.

- UNIT I** Data base System Applications, data base System VS file System – View of Data – Data Abstraction – Instances and Schemas – data Models – the ER Model – Relational Model – Other Models – Database Languages – DDL – DML – database Access for applications Programs – data base Users and Administrator – Transaction Management – data base System Structure – Storage Manager – the Query Processor. History of Data base Systems. Data base design and ER diagrams – Beyond ER Design Entities, Attributes and Entity sets – Relationships and Relationship sets – Additional features of ER Model – Concept Design with the ER Model – Conceptual Design for Large enterprises.
- UNIT II** Introduction to the Relational Model – Integrity Constraint Over relations – Enforcing Integrity constraints – Querying relational data – Logical data base Design – Introduction to Views – Destroying altering Tables and Views. Relational Algebra – Selection and projection set operations – renaming – Joins – Division – Examples of Algebra overviews – Relational calculus – Tuple relational Calculus – Domain relational calculus – Expressive Power of Algebra and Calculus
- UNIT III** Form of Basic SQL Query – Examples of Basic SQL Queries – Introduction to Nested Queries – Correlated Nested Queries Set – Comparison Operators – Aggregative Operators – NULL values – Comparison using Null values – Logical connectivity's – AND, OR and NOT – Impact on SQL Constructs – Outer Joins – Disallowing NULL values – Complex Integrity Constraints in SQL Triggers and Active Data bases.
- UNIT IV** Schema refinement – Problems Caused by redundancy – Decompositions – Problem related to decomposition – reasoning about FDS – FIRST, SECOND, THIRD Normal forms – BCNF – Lossless join Decomposition – Dependency preserving Decomposition – Schema refinement in Data base Design – Multi valued Dependencies – FORTH Normal Form  
Transaction Concept- Transaction State- Implementation of Atomicity and Durability – Concurrent – Executions – Serializability Recoverability – Implementation of Isolation – Testing for serializability- Lock – Based Protocols – Timestamp Based Protocols Validation- Based Protocols – Multiple Granularity
- UNIT V** Recovery and Atomicity – Log – Based Recovery – Recovery with Concurrent Transactions – Buffer Management – Failure with loss of nonvolatile storage-Advance Recovery systems- Remote Backup systems Data on External Storage – File Organization and Indexing – Cluster Indexes, Primary and Secondary Indexes – Index data Structures – Hash Based Indexing – Tree base Indexing – Comparison of File Organizations – Indexes and Performance Tuning- Intuitions for tree Indexes – Indexed Sequential Access Methods (ISAM) – B+ Trees: A Dynamic Index Structure.

## TEXT BOOKS:

1. Data base System Concepts, Silberschatz, Korth, McGraw Hill, 5<sup>th</sup> edition.
2. Data base Management Systems, Raghurama Krishnan, Johannes Gehrke, TATA McGraw Hill 3rd Edition

## REFERENCES:

1. Data base Systems design, Implementation, and Management, Peter Rob & Carlos Coronel 7th Edition.
2. Fundamentals of Database Systems, Elmasri Navrate Pearson Education
3. Introduction to Database Systems, C.J. Date Pearson Education

# Chhattisgarh Swami Vivekanand Technical University, Bhilai

Branch: **Computer Science & Engineering**  
Subject: **Microprocessor & Interfaces Laboratory**  
Total Lab Periods: **36**  
Maximum Marks: **40**

Semester: **V**  
Code: **322561(22)**  
Batch Size: **30**  
Minimum Marks: **20**

## List of Experiment to be performed

1. To perform addition & subtraction of two 8 – bit hexadecimal numbers.
2. To perform addition & subtraction 16 – bit hexadecimal numbers.
3. To perform addition & subtraction 32 – bit hexadecimal numbers.
4. To perform addition & subtraction of two 8 – bit decimal numbers and store the result in DX register.
5. To perform addition & subtraction of two decimal digits 9 and 7 using ASCII code store the result in ASCII format.
6. To perform addition & subtraction of two decimal digits 97 and 25 using ASCII code store the result in ASCII format in CX-BX register.
7. To perform multiplication of 4 and 5 .
8. To perform division of 16 – bit number with 8-bit number.
9. To perform multiplication of two 8-bit numbers using ASCII code store the result in ASCII form in DX.
10. To perform division of two 8-bit numbers using ASCII code store the result in ASCII form in DX register.
11. To solve Arithmetic equation  $3AX+5DX+BP$  and store the result in CX register.
12. To solve Arithmetic equation  $(P*Q)+(R*S)$ .
13. To add only positive number from 100 data bytes.
14. To write a program to add series of 20 bytes.
15. To find positive & negative byte from 100 data bytes.
16. To find largest & smallest byte from block of data.

- List of Equipment's/Machine Required: 8086 based microprocessor kit, MASM assembler, 8086 simulator, PCs.

## Reference Books:

1. IBM PC Assembly Language and Programming, P. Abel, 5th Edition, PHI/Pearson Education.
2. Introduction To Assembly Language Programming, SivaramaP.Dandamudi, Springer Int. Edition,2003.
3. The 8088 and 8086 Microprocessors: Programming , Interfacing,Software,Hardware and Application,4th edition,W.A.Triebel,A.Singh,N.K.Srinath,Pearson Education

# Chhattisgarh Swami Vivekanand Technical University, Bhilai

Branch: **Computer Science & Engineering**  
Subject: **Programming in Java Laboratory**  
Total Lab Periods: **36**  
Maximum Marks: **40**

Semester: **V**  
Code: **322562(22)**  
Batch Size: **30**  
Minimum Marks: **20**

## List of Experiment to be performed

1. Write a program to check whether a number is an Armstrong number or not.
2. Write a program to sort a stream of Strings.
3. Write a program to perform multiplication of two matrices.
4. Write a program to find the volume of a box having its side w, h, d means width, height and depth. Its volume is  $v=w*h*d$  and also find the surface area given by the formula  $s=2(w*h+h*d+d*w)$ , use appropriate constructors for the above.
5. Develop a program to illustrate a copy constructor so that a string may be duplicated into another variable either by assignment or copying.
6. Create a base class called shape. It contains two methods getxyvalue() and showxyvalue() for accepting co-ordinates and to display the same. Create the subclass called Rectangle which contains a method to display the length and breadth of the rectangle called showxyvalue(). Use overriding concept.
7. Write a program that creates an abstract class called dimension, creates two subclasses, rectangle and triangle. Include appropriate methods for both the subclass that calculate and display the area of the rectangle and triangle.
8. Write a program which throws Arithmetic Exception. Note the output; write another class (in a different file) that handles the Exception.
9. Create a user defined Exception class which throws Exception when the user inputs the marks greater than 100.
10. Write a program in which a Mythread class is created by extending the Thread class. In another class, create objects of the Mythread class and run them. In the run method print "CSV TU" 10 times. Identify each thread by setting the name.
11. Write a program using InetAddress class and also show the utility of URL and URL Connection classes.
12. Write a program which illustrates capturing of Mouse Events. Use Applet class for this.
13. Write a program using RMI in which a simple remote method is implemented.
14. Write a servlet program using HttpServlet class. Also give the appropriate HTML file which posts data to the servlet.
15. Write a JDBC program for Student Mark List Processing.
16. Design a text editor which is having some of the features of notepad.

## Reference Books:

1. Java complete reference - Naughton schildt (TMH)
2. Java programming – E Balagurusamy
3. Java 2 Black book – Steven Holzner
4. Java Examples in a nutshell – O' Reilly

# Chhattisgarh Swami Vivekanand Technical University, Bhilai

Branch: **Computer Science & Engineering**  
Subject: **UNIX & Shell Programming Laboratory**  
Total Lab Periods: **36**  
Maximum Marks: **40**

Semester: **V**  
Code: **322563(22)**  
Batch Size: **30**  
Minimum Marks: **20**

**Note: Use Bash for Shell scripts.**

## List of Experiment to be performed

1. a) Write a shell script that accepts a file name, starting and ending line numbers as arguments and displays all the lines between the given line numbers.  
b) Write a shell script that deletes all lines containing a specified word in one or more files supplied as arguments to it.
2. a) Write a shell script that displays a list of all the files in the current directory to which the user has read, write and execute permissions.  
b) Write a shell script that receives any number of file names as arguments checks if every argument supplied is a file or a directory and reports accordingly. Whenever the argument is a file, the number of lines on it is also reported.
3. a) Write a shell script to list all of the directory files in a directory.  
b) Write a shell script to find factorial of a given integer.
4. a) Write an awk script to count the number of lines in a file that do not contain vowels.  
b) Write an awk script to find the number of characters, words and lines in a file.
5. a) Write a shell script that accepts a list of file names as its arguments, counts and reports the occurrence of each word that is present in the first argument file on other argument files.  
b) Write a c program that makes a copy of a file using standard I/O and system calls.
6. a) Implement in C the following Unix commands using System calls  
    o cat, ls, mv  
b) Write a C program to emulate the Unix ls -l command.
7. Write a program that takes one or more file/directory names as command line input and reports the following information on the file.  
a) File type.  
b) Number of links.  
c) Time of last access.  
d) Read, Write and Execute permissions.
8. a) Write a C program to list for every file in a directory, its inode number and file name.  
b) Write a C program that demonstrates redirection of standard output to a file. Ex: ls > fl.
9. a) Write a C program to create a child process and allow the parent to display "parent" and the child to display "child" on the screen.  
b) Write a C program that illustrates how to execute two commands concurrently with a command pipe. Ex:- ls -l | sort
10. a) Write a C program to create a Zombie process.  
b) Write a C program that illustrates how an orphan is created.
11. a) Write C programs that illustrate communication between two unrelated processes using named pipe.  
b) Write a C program (sender.c) to create a message queue with read and write permissions to write 3 messages to it with different priority numbers.  
c) Write a C program (receiver.c) that receives the messages (from the above message queue as specified in (21)) and displays them.
12. a) Write a shell script to accept three numbers and display the largest.  
b) Write a shell script to find the number of files in a directory.  
c) Write a shell script to display first ten positive numbers using until loop.
13. a) Write a shell script to check if a particular user has logged in or not. If not, continue the loop till he/she logins. Once the required user logins, display a message.  
b) Write a shell script to accept the name, grade, and basic salary from the user. Write the details into a file called employee, separating the fields with a colon (.) continue the process till the user wants.
14. a) Write a shell script to check whether a file is existing or not.  
b) Write a shell script to find the mode of a file in a directory.  
c) Write a shell script which will accept different numbers and find their sum.
15. a) Write a menu driven program to display a menu of options and depending upon the user's choice execute the associated command.  
b) Write a shell script to calculate the total salary payable to all the employees from the employee file. The salary should be taken from the 8th field of the employee file.
16. a) Write a shell script to copy the source file to the target file.  
b) Write a shell script to print the first 10 odd numbers using the while loop.  
c) Write a shell script to reverse the digits of a given number.

## References:

1. Unix Shell programming, Yashwanth Kanitkar, 1st Edition, BPB Publisher
2. Unix for programmers and users, 3rd edition, Graham Glass, King Ables, Pearson education.
3. Unix programming environment, Kernighan and Pike, PHI. / Pearson Education
4. The Complete Reference Unix, Rosen, Host, Klee, Farber, Rosinski, Second Edition, TMH.

# Chhattisgarh Swami Vivekanand Technical University, Bhilai

Branch: **Computer Science & Engineering**  
Subject: **Data Base Management System Laboratory**  
Total Lab Periods: **36**  
Maximum Marks: **40**

Semester: **V**  
Code: **322564(22)**  
Batch Size: **30**  
Minimum Marks: **20**

## H/W Requirement:

- At least Dual Core or Core-I3 Pro Computing System, 2GB RAM, 80GB HDD

## Software

- All systems are configured in **DUAL BOOT** mode i.e., Students can boot from Windows 7/8 or Linux as per their lab requirement.
- MySQL/PgSQL/Oracle May be used as Database Management System
- Systems are provided for students in the 1:1 ratio.

## LIST OF EXPERIMENTS

### 1 Database Schema for a customer-sale scenario

Customer(**Cust id : integer**, cust\_name: string)

Item(**item\_id: integer**, item\_name: string, price: integer)

Sale(**bill\_no: integer**, bill\_data: date, **cust\_id: integer**, **item\_id: integer**, qty\_sold: integer)

For the above schema, perform the following—

- Create the tables with the appropriate integrity constraints
- Insert around 10 records in each of the tables
- List all the bills for the current date with the customer names and item numbers
- List the total Bill details with the quantity sold, price of the item and the final amount
- List the details of the customer who have bought a product which has a price>200
- Give a count of how many products have been bought by each customer
- Give a list of products bought by a customer having cust\_id as 5
- List the item details which are sold as of today
- Create a view which lists out the bill\_no, bill\_date, cust\_id, item\_id, price, qty\_sold, amount
- Create a view which lists the daily sales date wise for the last one week

### 2 Database Schema for a Student Library scenario

Student(**Stud\_no : integer**, Stud\_name: string)

Membership(**Mem\_no: integer**, **Stud\_no: integer**)

Book(**book\_no: integer**, book\_name:string, author: string)

Iss\_rec(**iss\_no:integer**, iss\_date: date, **Mem\_no: integer**, **book\_no: integer**)

For the above schema, perform the following—

- Create the tables with the appropriate integrity constraints
- Insert around 10 records in each of the tables
- List all the student names with their membership numbers
- List all the issues for the current date with student and Book names
- List the details of students who borrowed book whose author is CJDATE
- Give a count of how many books have been bought by each student
- Give a list of books taken by student with stud\_no as 5
- List the book details which are issued as of today
- Create a view which lists out the iss\_no, iss\_date, stud\_name, book name
- Create a view which lists the daily issues-date wise for the last one week

### 3 Database Schema for a Employee-pay scenario

employee(**emp\_id : integer**, emp\_name: string)

department(**dept\_id: integer**,dept\_name:string)

paydetails(**emp\_id : integer**, **dept\_id: integer**, basic: integer, deductions: integer, additions: integer, DOJ: date)

payroll(**emp\_id : integer**, pay\_date: date)

For the above schema, perform the following—

- Create the tables with the appropriate integrity constraints
- Insert around 10 records in each of the tables
- List the employee details department wise
- List all the employee names who joined after particular date
- List the details of employees whose basic salary is between 10,000 and 20,000
- Give a count of how many employees are working in each department
- Give a names of the employees whose netsalary>10,000
- List the details for an employee\_id=5
- Create a view which lists out the emp\_name, department, basic, dedeuctions, netsalary
- Create a view which lists the emp\_name and his netsalary

### 4 Database Schema for a Video Library scenario

Customer(cust\_no: integer,cust\_name: string)

Membership(**Mem\_no: integer**, **cust\_no: integer**)

Cassette(**cass\_no:integer**, cass\_name:string, Language: String)

**Name of the Programme: Bachelor of Engineering ::::: Duration of the Programme: Four years**

Iss\_rec(iss\_no: integer, iss\_date: date, mem\_no: integer, cass\_no: integer)

For the above schema, perform the following—

- a) Create the tables with the appropriate integrity constraints
- b) Insert around 10 records in each of the tables
- c) List all the customer names with their membership numbers
- d) List all the issues for the current date with the customer names and cassette names
- e) List the details of the customer who has borrowed the cassette whose title is “ The Legend”
- f) Give a count of how many cassettes have been borrowed by each customer
- g) Give a list of book which has been taken by the student with mem\_no as 5
- h) List the cassettes issues for today
- i) Create a view which lists out the iss\_no, iss\_date, cust\_name, cass\_name
- j) Create a view which lists issues-date wise for the last one week

**5 Database Schema for a student-Lab scenario**

Student(stud\_no: integer, stud\_name: string, class: string)

Class(class: string, descrip: string)

Lab(mach\_no: integer, Lab\_no: integer, description: String)

Allotment(Stud\_no: Integer, mach\_no: integer, dayof week: string)

For the above schema, perform the following—

- a) Create the tables with the appropriate integrity constraints
  - b) Insert around 10 records in each of the tables
  - c) List all the machine allotments with the student names, lab and machine numbers
  - d) List the total number of lab allotments day wise
  - e) Give a count of how many machines have been allocated to the ‘CSIT’ class
  - f) Give a machine allotment details of the stud\_no 5 with his personal and class details
  - g) Count for how many machines have been allocated in **Lab\_no 1** for the day of the week as “Monday”
  - h) How many students class wise have allocated machines in the labs
  - i) Create a view which lists out the stud\_no, stud\_name, mach\_no, lab\_no, dayofweek
  - j) Create a view which lists the machine allotment details for “Thursday”.
- 6 a) Write and execute subprogram to find largest number from the given three numbers.  
b) Write and execute subprogram using loop, while and for iterative control statement.
  7. a) Write and execute subprogram to check whether the given number is Armstrong or not  
b) Write and execute subprogram to generate all prime numbers below 100.
  - a) Write and execute subprogram to demonstrate the GOTO statement.
  - 8 b) Write a subprogram to demonstrate %type and %rowtype attributes
  - 9 a) Write and execute subprogram to demonstrate predefined exceptions  
b) Write and execute subprogram to demonstrate user defined exceptions
  - 10 a) Create a cursor, which displays all employee numbers and names from the EMP table.  
b) Create a cursor, which update the salaries of all employees as per the given data.
  - 11 a) Create a cursor, which displays names of employees having salary > 50000.  
b) Create a procedure to find reverse of a given number
  - 12 a) Create a procedure to update the salaries of all employees as per the given data  
b) Create a procedure to demonstrate IN, OUT and INOUT parameters
  - 13 a) Create a function to check whether given string is palindrome or not.  
b) Create a function to find sum of salaries of all employees working in depart number 10.
  - 14 a) Create a trigger before/after update on employee table for each row/statement.  
b) Create a trigger before/after delete on employee table for each row/statement.  
c) Create a trigger before/after insert on employee table for each row/statement.
  - 16 a) Create a Form to display employee details using SQL  
b) Create a Report to generate all employee annual salaries....

**Additional Programs**

- 1 Create a form using Forms 6i to display Employee table data.
- 2 Create a Master/details relationship form which perform Add New, Search, Delete, Save and Update on the records
- 3 Generate a report to calculate employee’s salaries department wise from employee table.
- 4 Create a Report to generate the details of employee table including sum and average salaries department wise.

**Course Outcome:** Upon successful completion of this Lab the student will be able to:

- Creating , Modify database Tables and manipulate data in the Tables
- Performing database operations in a procedural manner using pl/sql
- Performing database operations (create, update, modify, retrieve, etc..) using front-end tools like D2K/Gambas/Php/Servlet/JSP.
- Design and Develop applications like banking, reservation system, etc.
- Design and implement a small database project using RDBMS to understand the concept of a database transaction and related database facilities.

# Chhattisgarh Swami Vivekanand Technical University, Bhilai

Name of Program:	<b>Bachelor of Engineering</b>	Semester:	<b>V</b>
Branch:	<b>Common to All Branches</b>	Code:	<b>300565 (46)</b>
Subject:	<b>Personality Development</b>	Tutorial Period:	<b>NIL</b>
No. of Lectures:	<b>2/Week</b>	Marks in TA:	<b>20</b>
Total Marks in ESE:	<b>NIL</b>		
Minimum number of Class Tests to be conducted: <b>Two</b>			

**Objective:** The course is introduced to develop one's outer and inner personality tremendously and enrich the abilities to enable one to meet the challenges associated with different job levels. Personality Development is essential for overall development of an individual apart from gaining technical knowledge in the subject.

## Course Objectives

Upon completion of this course, the student shall be able

- To understand the concept of personality and image;
- To develop leadership, listening and interacting skills;
- To develop attitudinal changes;
- To develop decision-making qualities; and
- To communication skill.

- UNIT I Personality concepts:** What is Personality – its physical and psychic aspects. How to develop a positive self-image. How to aim at Excellence. How to apply the cosmic laws that govern life and personality. How to improve Memory – How to develop successful learning skills. How to develop and effectively use one's creative power. How to apply the individual MOTIVATORS that make you a self-power personality.
- UNIT II Interpersonal Skills:** Leadership: Leaders who make a difference, Leadership: your idea, What do we know about leadership? If you are serious about Excellence. Concepts of leadership, Two important keys to effective leadership, Principles of leadership, Factors of leadership, Attributes. Listening: Listening skills, How to listen, Saying a lot- just by listening, The words and the music, How to talk to a disturbed person, Listening and sometimes challenging. How to win friends and influence people, How to get along with others. How to develop art of convincing others. How can one make the difference. How to deal with others particularly elders. Conflicts and cooperation.
- UNIT III Attitudinal Changes: Meaning of attitude,** benefits of positive attitudes, How to develop the habit of positive thinking.  
**Negative attitude and wining:** What is FEAR and how to win it. How to win loneliness. How to win over FAILURE. How to win over PAIN. How to win over one's ANGER and others anger. What is stress and how to cope up with it? The art of self-motivation. How to acquire mental well-being. How to acquire physical well-being.
- UNIT IV Decision Making:** How to make your own LUCK. How to plan goals/objectives and action plan to achieve them. How to make RIGHT DECISION and overcome problems. How to make a Decision. Decision making: A question of style. Which style, when? People decisions: The key decisions. What do we know about group decision making? General aids towards improving group decision making.
- UNIT V Communication Skills: Public Speaking:** Importance of Public speaking for professionals. The art of Speaking - Forget the fear of presentation, Symptoms of stage fear, Main reason for speech failure, Stop failures by acquiring Information; Preparation & designing of speech, Skills to impress in public speaking & Conversation, Use of presentation aids & media.  
**Study & Examination:** How to tackle examination, How to develop successful study skills.  
**Group discussions:** Purpose of GD, What factors contribute to group worthiness, Roles to be played in GD.

## Course Outcomes:

- The students will be able to develop inner and outer personality exposure;
- The students will be able to develop effective leadership qualities and interacting skills;
- The students will be able to develop positive attitude, motivating skills and develop winning philosophies;
- The students will be able to develop decision-making tools; and
- The students will be able to develop group presentation, public speaking and impressive conversation.

## Text Books:

1. Basic Managerial Skills for all by E. H. McGrawth, prentice Hall India Pvt. Ltd., 2006
2. Basic Employability Skills by P. B. Deshmukh, BSP Books Pvt. Ltd., Hyderabad, 2014

## Reference Books:

1. How to Develop a Pleasing Personality by Atul John Rego, Better Yourself Books, Mumbai, 2000
2. How to Succeed by Brain Adams, Better Yourself Books, Mumbai, 1969
3. Personality: Classic Theories & Modern Research; Friedman ; Pearson Education, 2006
4. How to Win Friends and Influence People by Dale Carnigie, A. H. Wheeler 2006