Walchand College of Engineering, Sangli

(Government Aided Autonomous Institute)

AY 2025-26

Course Information	
Programme	B.Tech. (Computer Science and Engineering)
Class, Semester	Third Year B. Tech., Sem V
Course Code	7CS301
Course Name	Compiler Design
Desired Requisites:	Formal Language and Automata Theory, Discrete Mathematics

Teaching Scheme		**Examination Scheme (Marks)**			
Lecture	3 Hrs/week	**ISE**	**MSE**	**ESE**	**Total**
Tutorial	-	20	30	50	100
Practical	-				
Interaction	-	**Credits: 3**			

Course Objectives	
1	To introduce fundamentals of compiler design and various tools used to design a compiler
2	To inculcate role of various phases involved during design of a compiler and impart in depth working of each phase
3	To impart the design of various phases of a compiler using compiler design tools and techniques

Course Outcomes (CO) with Bloom's Taxonomy Level		
CO1	Discuss the need of a compiler, fundamental concepts and various tools used to design a compiler.	Understanding
CO2	Demonstrate the role and working of each phase involved during compilation process.	Applying
CO3	Compare and analyze the working of various phases of compiler	Analyzing
CO4	Verify and assess the working of various phases involved in design of a Compiler	Evaluating

Module	**Module Contents**	**Hours**
I	**Module 1: Fundamentals of Compiler** Overview- Structure of a compiler, applications of compiler, one pass and two pass compiler. **Lexical analysis** - The role of a lexical analyzer, specification of tokens, recognition of tokens, LEX.	6
II	**Module 2 Syntax Analysis** Context-free grammar, writing grammars for context free environments, parse trees and ambiguity, role of parser, specification and recognition of tokens, top-down parsing, recursive descent and predictive parsers (LL), bottom-up parsing, operator precedence parsing, LR, SLR and LALR parsers.	9
III	**Module 3 Syntax Directed Translation & Run time environments** Syntax-directed definitions, evaluation orders for attributes of an SDD, S-attributed and L-attributed SDDs, construction of syntax tree, source language issues, storage organization and allocation strategies, parameter passing, symbol table organizations and generations, dynamic storage allocations.	6

IV	**Module 4 Intermediate Code Generation** Intermediate languages, declarations, different intermediate representations –quadruples, triples, trees, flow graphs, SSA forms, and their uses; assignment statements and Boolean expressions, case statements, back patching, procedure calls.	6
V	**Module 5 Code Optimization** Sources of optimization, basic blocks and flow graphs, optimization of basic blocks, loops in flow graphs, loop optimization, machine-independent optimization, machine-dependent optimization, dead-code Elimination, code improving transformations.	6
VI	**Module 6 Code Generation** Issues in the design of a code generator, run time storage management; simple code generator- register and address descriptors, code generation algorithm, design of the function getReg, DAG, peephole optimization, register allocation and assignment, selection of instruction, register allocation, parallel compilation, Just-in-Time compiler, study of compiler construction tools.	6

Text Books	
1	A.V. Aho, R. Shethi and J.D. Ullman, “Compilers - Principles, Techniques and Tools”, Pearson Education, Second Edition, 2007.
2	D.M. Dhamdhere, “Systems Programming and Operating Systems”, Tata McGraw- Hill Publishing Company limited, New Delhi, Second revised Edition, 2005.

References	
1	K Cooper, L Torczon, “Engineering a Compiler”, Morgan Kaufmann, Second Edition, 2011
2	John J Donavan, “System Programming”, Tata McGraw- Hill Publishing Company limited, New Delhi
3	Sumitabha Das, “Unix Concepts and Administration”, TMGH, 3rd Edition
4	A.V. Aho, R. Shethiand J.D. Ullman, “Compilers - Principles, Techniques and Tools”, Addison Wesley Publishing Company, 2007

Useful Links	
1	https://onlinecourses.nptel.ac.in/noc21_cs07/preview
2	https://nptel.ac.in/courses/106108052

CO-PO Mapping

	Programme Outcomes (PO)												**PSO**	
	1	2	3	4	5	6	7	8	9	10	11	12	1	2
CO1	2												2	
CO2	3	2											3	
CO3		2	3											
CO4		2	3		1				1				3	

The strength of mapping is to be written as 1,2,3; Where, 1:Low, 2:Medium, 3:High
Each CO of the course must map to at least one PO.

Assessment

The assessment is based on MSE, ISE and ESE. MSE shall be typically on modules 1 to 3. ISE shall be taken throughout the semester in the form of teacher’s assessment. Mode of assessment can be field visit, assignments etc. and is expected to map at least one higher order PO. ESE shall be on all modules with around 40% weightage on modules 1 to 3 and 60% weightage on modules 4 to 6. For passing a theory course, Min. 40% marks in (MSE+ISE+ESE) are needed and Min. 40% marks in ESE are needed. (ESE shall be a separate head of passing)

Walchand College of Engineering, Sangli

(Government Aided Autonomous Institute)

AY 2025-26

Course Information

Programme	B.Tech. (Computer Science Engineering)
Class, Semester	Third Year B. Tech., Sem V
Course Code	7CS352
Course Name	Cutting Edge Technologies Lab
Desired Requisites:	Basics of programming

Teaching Scheme		**Examination Scheme (Marks)**			
Practical	2 Hrs/ Week	**LA1**	**LA2**	**Lab ESE**	**Total**
Lecture	Hrs/ Week	30	30	40	100
		Credits: 1			

Course Objectives

1	To provide hands-on experience and practical implementation skills in one of the cutting-edge technologies. Currently following are the specialized domains: 1. iOS App Development 2. Robotics 3. Parallel Programming.
2	Students will choose one lab based on interest or departmental allocation.
3	To enhance domain-specific skills and prepare students for industry-relevant projects.

Course Outcomes (CO) with Bloom's Taxonomy Level

At the end of the course, the students will be able to,

CO	**Course Outcome Statement/s**	**Bloom's Taxonomy Level**	**Bloom's Taxonomy Description**
CO1	illustrate the fundamental concepts and apply domain-specific tools and techniques to solve real-world problems.	III	Applying
CO2	demonstrate technical proficiency by implementing standard solutions	IV	Analysing
CO3	select appropriate components/methods to solve real-world problems.	V	Evaluating
CO4	build an application, individually or in a team for solving real-world problems.	VI	Creating

1. iOS Application Development

List of Experiments / Lab Activities/Topics

List of Lab Activities:

1. Introduction to swift and playground.
2. Programs based on data types, control statements and operators.
3. Programs based on Functions, Strings.
4. Programs based on Structures, Classes and Optional.
5. Controls in Action, Auto Layout and Stack Views.
6. Segues, Navigation and Tab bar Controllers.
7. Type Casting and Inspection, Guard, Constant and Variable Scope, Enumerations.
8. Application design cycle, iterate over the design, create a prototype.
9. Data collection using swift.
10. Connecting the app to the database.

Textbooks

1	Develop in swift fundamentals – Apple Education
2	Develop in swift Data Collections - Apple Education
3	Neil Smyth, "Android Studio 3.6 Development Essentials - Java Edition: Developing Android 10 (Q) Apps Using Android Studio 3.6, Java and Android Jetpack", Payload Media, 2020, ISBN-13: 978-1951442156

References

1	Develop in swift fundamentals notes

2	Best Book for Step-by-step Learners: Swift: A Step-by-Step Guide for Absolute Beginners by Daniel Bell
3	Dawn Griffiths, David Griffiths, "Head First Android Development", O'Reilly Media, 2nd Edition, 2017, ISBN: 9781491974056
Useful Links	
1	https://docs.swift.org/swift-book/documentation/the-swift-programming-language/
2	https://developer.android.com/docs

CO-PO Mapping

	Programme Outcomes (PO)												PSO	
	1	2	3	4	5	6	7	8	9	10	11	12	1	2
CO1				2	2								1	2
CO2				2	2								1	1
CO3				3	2								1	1
CO4				2	2			2	2	2	1	1	1	2

The strength of mapping is to be written as 1,2,3; where, 1: Low, 2: Medium, 3: High
Each CO of the course must map to at least one PO, and preferably to only one PO.

2. Robotics

List of Experiments / Lab Activities/Topics

List of Lab Activities:

1. Introduction to robotics
2. Installation of dubot studio
3. Implementation of suction cup end effector with
4. Dubot studio teach and play back application
5. Implementation of suction cup end effector with
6. Blockly and script application
7. Implementation of gripper end effector with teach & play back, blockly and script application
8. Implementation of dobot magician
9. Implementation pen end effector of dobot magician
10. Study of blender a 3d animation tool
11. Generate api to connect and move dobot arm from
12. One location to another using python
13. Implementation of blender
14. Study of intel depth realsense camera.
15. Implementation of intel depth realsense camera with assigned tasks
16. Study and implementation of rtk module
17. Study and implementation of lidars
18. Study and implementation of turtlebot

Textbooks	
1	**Introduction to Robotics: Mechanics and Control** – *John J. Craig, Pearson (4th Edition)*
2	**Learning ROS for Robotics Programming** – *Aaron Martinez, Enrique Fernandez, Packt Publishing (2nd Edition)*
3	**Python Robotics Projects** – *Joseph Howse, Packt Publishing*
References	
1	http://wiki.ros.org/
2	https://www.dobot-robots.com/
3	http://emanual.robotis.com/docs/en/platform/turtlebot3/overview/
Useful Links	
1	https://www.youtube.com/playlist?list=PLjEaoINr3zgFX8ZsChQVQsuDSjEqdWMAD
2	https://oceanservice.noaa.gov/facts/lidar.html#:~:text=Lidar%2C%20which%20stands%20for%20Light,variable%20distances)%20to%20the%20Earth.

3	https://www.intelrealsense.com/depth-camera-d435/

CO-PO Mapping

	Programme Outcomes (PO)												PSO	
	1	2	3	4	5	6	7	8	9	10	11	12	1	2
CO1	3	2	1	-	-	-	-	-	-	-	-	-	-	-
CO2	2	3	2	-	2	-	-	-	-	-	-	-	-	-
CO3	2	3	2	1	3	-	-	-	-	-	-	-	-	-
CO4	1	3	1	2	-	-	2	2	2	-	-	-	-	-

The strength of mapping is to be written as 1,2,3; where, 1: Low, 2: Medium, 3: High
Each CO of the course must map to at least one PO, and preferably to only one PO.

3. Parallel Programming

List of Experiments / Lab Activities/Topics

List of Lab Activities:

1. Introduction to parallel programming, OpenMP installation, simple “Hello World” programs.
2. Parallel region creation, thread identification using OpenMP.
3. Parallelizing loops with #pragma omp parallel for, and applying scheduling clauses.
4. Use of private, shared, firstprivate, lastprivate clauses in OpenMP.
5. Synchronization constructs: critical, barrier, atomic, ordered, nowait.
6. Reduction operations and nested loop parallelism using collapse.
7. Introduction to MPI: simple point-to-point communication using MPI_Send and MPI_Recv.
8. Collective communication: broadcast, scatter, gather, reduce in MPI.
9. Matrix-vector multiplication using MPI with row-wise distribution.
10. Comparison of OpenMP and MPI versions of the same problem (e.g., sum of array elements).
11. Mini Project Phase I: Choose a problem and plan hybrid or comparative OpenMP/MPI solution.
12. Mini Project Phase II: Final implementation and Performance Analysis.

Textbooks

1	Quinn, M. J. “Parallel Programming in C with MPI and OpenMP”, 1st Edition, Tata McGraw Hill Education, 2003.
2	Pacheco, P. “An Introduction to Parallel Programming”, 1st Edition, Morgan Kaufmann, 2011.
3	Grama, A., Gupta, A., Karypis, G., & Kumar, V. “Introduction to Parallel Computing”, 2nd Edition, Pearson Education, 2003.

References

1	Gropp, W., Lusk, E., & Skjellum, A. “Using MPI: Portable Parallel Programming with the Message-Passing Interface”, 2nd Edition, MIT Press, 1999.
2	Snu, V. S. “Parallel Programming: Concepts and Practice”, PHI Learning, 2020.
3	Chandra, R., Menon, R., Dagum, L., Kohr, D., Maydan, D., & McDonald, J. “Parallel Programming in OpenMP”, Morgan Kaufmann, 2001.

Useful Links

1	https://mpitutorial.com
2	https://openmp.org

CO-PO Mapping

	Programme Outcomes (PO)												PSO	
	1	2	3	4	5	6	7	8	9	10	11	12	1	2
CO1	3		2		3								3	2
CO2	3		3		3								3	2
CO3	2	3		3									2	3
CO4	2			3									2	3

The strength of mapping is to be written as 1,2,3; where, 1: Low, 2: Medium, 3: High
Each CO of the course must map to at least one PO, and preferably to only one PO.