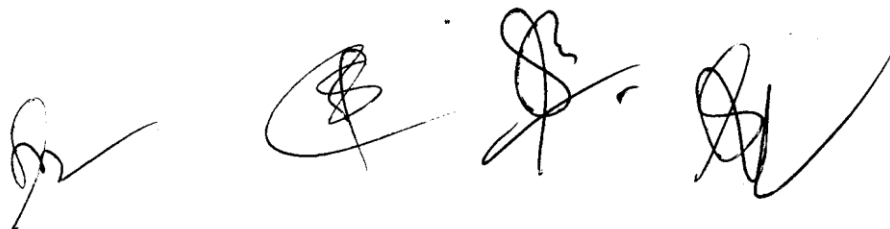


**ST. ALOYSIUS COLLEGE (AUTONOMOUS), JABALPUR****PART A: Introduction**

Program: Certificate	Class: B.Sc.	Year: I Year (sem 1)	SESSION 2023-24
Subject: Computer Science			
1.	Course Code	SI-COSC IT	
2.	Course Title	Computer System Architecture ( Paper 1)	
3.	Course Type (Elective/Generic Elective)	Elective Course	
4.	Pre-Requisite (if any)	To study this course, a student must have had the subject Physics/ Math in 12th class.	
5.	Course Outcomes(CO)	<p>On completion of this course, learners will be able to:</p> <p>CO1. Understand the basic structure, operation and characteristics of digital computer.</p> <p>CO2. Be able to design simple combinational digital circuits based on given parameters.</p> <p>CO3. Familiarity with working of arithmetic and logic unit as well as the concept of pipelining.</p> <p>CO4. Know about hierarchical memory system including cache memories and virtual memory.</p> <p>CO5. Understand concept and advantages of parallelism, threading, multi-processors and multi-core processors.</p> <p>Know the contributions of Indians in the field of computer architecture and related technologies.</p>	
6.	Credit Value	Theory 3 Credits	
7.	Total Marks	Max. Marks : 100	Min. Passing Marks: 35



PART B: Content of the Course		
Module	Topics	No. of Lectures
I	Fundamentals of Digital Electronics: Number System, Conversions, Binary Arithmetic, Complements, Fixed-Point Representation, Floating-Point Representation, Binary and other Codes, Error Detection Codes.	10
II	Logic Gates, Boolean Algebra, Map Simplification, K-Map, Combinational Circuits, Sequential Circuits, Simple Combinational circuit design problems.	10
III	Combinational Circuits- Adder, Subtractor, Multiplexer, De-multiplexer, Decoders, Encoders, Sequential Circuits - Flip - Flops, SR, D, T, JK,, Registers, Types of Registers, Counters, Types of Counters.	10
IV	Instructions, Instructions Formats, RISC, CISC, DMA Data Transfer, Auxiliary Memory, Cache Memory, Associative Memory, Virtual Memory, Flynn's classification - Introduction to SISD, SIMD, MISD, MIMD, Parallelism, Multicore processors.	10

Keywords/Tags: Digital Electronics, Logic Gates, Circuits, Instruction formats, Parallelism, Memory hierarchy, Multicore, Multi-threading, SISD, SIMD, MISD, MIMD.

#### PART C: Learning Resources

Textbooks, Reference Books, Other Resources

Suggested Readings:

- M. Morris Mano, "Computer System Architecture", PHI.
- Heuring Jordan , "Computer System Design & Architecture" (A.W.L.)
- William Stalling, "Computer Organization & Architecture", Pearson Education Asia.
- V. Carl Hamacher , "Computer Organization", TMH
- Tannenbaum, "Structured Computer Organization ", PHI.

#### PART D: Assessment and Evaluation

<b>Internal Assessment :</b> Continuous Comprehensive Evaluation (CCE) : 40 Marks		<b>External Assessment:</b> University Exam (UE) : 60 Marks	
Three test will be taken of which best of two marks will be considered		Time : 02.00 Hours	
Objective type Text I	20 Marks	Section (A) : Very short questions (1 from each unit)	1 x 5 = 5 Marks
Class Test II (Subjective)	20 Marks	Section (B) : 5 Short Questions (200 Words Each)	4 x 5 = 20 Marks
Class Test III (Subjective)	20 Marks		
		Section (C): 5 Long Questions (500	7 x 5 = 35

		Words Each)	Marks
Total	<b>40 Marks</b>	Total	<b>60 Marks</b>

**PART D: Content of the Course**

No. of Lab. Practical s (in hours per week): 2 Hrs. per week

Total No. of Labs:

	Suggestive list of Practical	No. of Labs.
	<ol style="list-style-type: none"> <li>1. To study basic gates (AND, OR, NOT) and verify their truth tables.</li> <li>2. To study and verify NAND as Universal gate using IC 7400.</li> <li>3. To realize basic gate AND from Universal gate NAND.</li> <li>4. To realize basic gate OR from Universal gate NAND.</li> <li>5. To realize basic gate NOT from Universal gate NAND.</li> <li>6. To study and verify NOR as Universal gate</li> <li>7. To realize basic gate AND from Universal gate NOR.</li> <li>8. To realize basic gate OR from Universal gate NOR.</li> <li>9. To realize basic gate NOT from Universal gate NOR.</li> <li>10. To study Half Adder using basic gates and verify its truth table.</li> <li>11. To study Full Adder using basic gates and verify its truth table.</li> <li>12. To design and construct RS flip Flop using gates and verifies the truth table.</li> <li>13. To design and construct JK Flip Flop using gates and verifies the truth table.</li> <li>14. To verify De-Morgan's First Law Theorem.</li> <li>15. To verify De-Morgan's Second Law Theorem.</li> </ol>	
	Keywords/Tags: Digital Electronics, Logic Gates, AND, OR, NOT, IC7486, IC 7400, NAND, NOR, IC 7483, Circuits, Flip Flop, De-Morgan's	

**PART D: Assessment and Evaluation**

<b>Internal Assessment : Continuous Comprehensive Evaluation (CCE) : 40 Marks</b>		<b>External Assessment: University Exam (UE) : 60 Marks</b>	
		Time : 02.00 Hours	
<b>Internal Assessment</b>	<b>Marks</b>	<b>External Assessment</b>	<b>Marks</b>
Lab Attendance	10 Marks	Practical record file	25 Marks
		Viva voce practical	10 Marks
Internal Viva	10 Marks	Execution	05 Marks
Practical File	20 Marks	Answer script	20 Marks
<b>Total</b>	<b>40 Marks</b>	<b>Total</b>	<b>60 Marks</b>

**ST. ALOYSIUS COLLEGE (AUTONOMOUS), JABALPUR****PART A: Introduction**

PROGRAM: Certificate		CLASS: B.Sc.	SEMESTER: I	SESSION 2023-24
Subject: Computer Science				
1.	Course Code	S1 - COSC IT		
2.	Course Title	Computer System Architecture (Paper I)		
3.	Course Type	Major/Minor		
4.	Pre-Requisite (if any)	To study this course, a student must have had the subject Physics/ Math in 12th class.		
5.	Course Learning Outcomes(CO)	On completion of this course, learners will be able to: CO1. Understand the basic structure, operation and characteristics of digital computer. CO2. Be able to design simple combinational digital circuits based on given parameters. CO3. Familiarity with working of arithmetic and logic unit as well as the concept of pipelining. CO4. Know about hierarchical memory system including cache memories and virtual memory. CO5. Understand concept and advantages of parallelism, threading, multi-processors and multi-core processors. Know the contributions of Indians in the field of computer architecture and related technologies.		
6.	Credit Value	Theory 4 Credits		
7.	Total Marks	Max. Marks : 100	Min. Passing Marks: 35	

**PART B: Content of the Course**

No. of Lectures (in hours per week): 2 Hrs. per week

Total No. of Lectures: 60 Hrs.

Module	Topics	No. of Lectures
I	Fundamentals of Digital Electronics: Number System-Binary, Decimal, Octal, Hexa-Decimal, Conversions, Binary Arithmetic-Addition, Subtraction, Multiplication, Division, Underflow, Overflow, Sign Magnitude, Complements-1's and 2's, Fixed-Point Representation, Floating-Point Representation.	10
II	Boolean Algebra, Reducing Boolean Expression, Logic Gates-AND, OR, NOT, Universal Gates-NAND, NOR, Analog and Digital Signals, Clock Waveform Timing, Map Simplification, K-Map- Two, Three and Four variables.	10

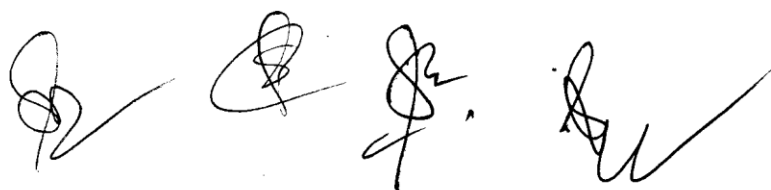


III	Combinational Circuits- Adder, Subtractor, Multiplexer, Demultiplexer, Decoders, Encoders, Binary Codes – Gray Codes, ASCII code, BCD code, EBCDIC, Error Detection Code and Correction Code, Hamming Code.	10
IV	Sequential Circuits - Flip - Flops, SR, D, T, JK, Master-Slave, Registers, Shift Registers- SISO, SIPO, PISO, PIPO, Counters, Instruction, Instruction Format, Instruction Codes, instructions Cycles, Addressing Modes.	10
V	Handshaking, Concepts of RISC, CISC, DMA Data Transfer, Auxiliary Memory, Cache Memory, Associative Memory, Virtual Memory, Flynn's classification - Introduction to SISD, SIMD, MISD, MIMD, Parallelism, Multicore processors.	10
Keywords/Tags: Digital Electronics, Logic Gates, Circuits, Instruction formats, Parallelism, Memory hierarchy, Multicore, Multi-threading, SISD, SIMD, MISD, MIMD.		

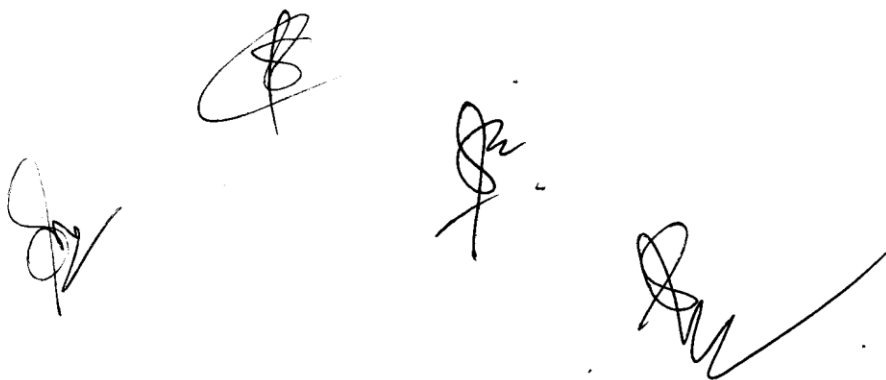
PART D: Assessment and Evaluation			
<b>Internal Assessment :</b> Continuous Comprehensive Evaluation (CCE) : 40 Marks Three test will be taken of which best of two marks will be considered		<b>External Assessment:</b> University Exam (UE) : 60 Marks Time : 02.00 Hours	
Objective type Text I	20 Marks	Section (A) : Very short questions (1 from each unit)	1 x 5 = 5 Marks
Class Test II (Subjective)	20 Marks	Section (B) : 5 Short Questions (200 Words Each)	4 x 5 = 20 Marks
Class Test III (Subjective)	20 Marks	Section (C): 5 Long Questions (500 Words Each)	7 x 5 = 35 Marks
Total	40 Marks	Total	60 Marks
Any remarks/suggestions: Focus of the course/teaching should be on developing ability of the student in analyzing a problem, building the logic and efficient code for the problem.			



PART D: Content of the Course		
No. of Lab. Practical s (in hours per week): 2 Hrs. per week		
Total No. of Labs: 15 Labs (30 HRS)		
	Suggestive list of Practical	No. of Labs.
	<ol style="list-style-type: none"> <li>1. To study basic gates (AND, OR, NOT) and verify their truth tables.</li> <li>2. To study and verify NAND as Universal gate using IC 7400.</li> <li>3. To realize basic gate AND from Universal gate NAND.</li> <li>4. To realize basic gate OR from Universal gate NAND.</li> <li>5. To realize basic gate NOT from Universal gate NAND.</li> <li>6. To study and verify NOR as Universal gate</li> <li>7. To realize basic gate AND from Universal gate NOR.</li> <li>8. To realize basic gate OR from Universal gate NOR.</li> <li>9. To realize basic gate NOT from Universal gate NOR.</li> <li>10. To study Half Adder using basic gates and verify its truth table.</li> <li>11. To study Full Adder using basic gates and verify its truth table.</li> <li>12. To design and construct RS flip Flop using gates and verifies the truth table.</li> <li>13. To design and construct JK Flip Flop using gates and verifies the truth table.</li> <li>14. To verify De-Morgan's First Law Theorem.</li> <li>15. To verify De-Morgan's Second Law Theorem.</li> <li>16. To study basic gates (AND, OR, NOT) and verify their truth tables.</li> <li>17. To study and verify NAND as Universal gate using IC 7400.</li> <li>18. To realize basic gate AND from Universal gate NAND.</li> <li>19. To realize basic gate OR from Universal gate NAND.</li> <li>20. To realize basic gate NOT from Universal gate NAND.</li> <li>21. To study and verify NOR as Universal gate</li> <li>22. To realize basic gate AND from Universal gate NOR.</li> <li>23. To realize basic gate OR from Universal gate NOR.</li> <li>24. To realize basic gate NOT from Universal gate NOR.</li> <li>25. To study Half Adder using basic gates and verify its truth table.</li> <li>26. To study Full Adder using basic gates and verify its truth table.</li> <li>27. To design and construct RS flip Flop using gates and verifies the truth table.</li> <li>28. To design and construct JK Flip Flop using gates and verifies the truth table.</li> <li>29. To verify De-Morgan's First Law Theorem.</li> <li>30. To verify De-Morgan's Second Law Theorem.</li> </ol>	15
	<p>Keywords/Tags:</p> <p>Digital Electronics, Logic Gates, AND, OR, NOT, IC7486, IC 7400, NAND, NOR, IC 7483. Circuits, Flip Flop, De-Morgan's Theorem.</p>	



PART D: Assessment and Evaluation			
<b>Internal Assessment : Continuous Comprehensive Evaluation (CCE) : 40 Marks</b>		<b>External Assessment: University Exam (UE) : 60 Marks</b> <b>Time : 02:00 Hours</b>	
<b>Internal Assessment</b>	<b>Marks</b>	<b>External Assessment</b>	<b>Marks</b>
Lab Attendance	10 Marks	Practical record file	25 Marks
		Viva voce practical	10 Marks
Internal Viva	10 Marks	Execution	05 Marks
Practical File	20 Marks	Answer script	20 Marks
<b>Total</b>	<b>40 Marks</b>	<b>Total</b>	<b>60 Marks</b>



**PART A: Introduction**

Program: <b>Certificate</b>		Class: <b>B.Sc</b>	Year: <b>I (sem 2)</b>	Session: <b>2023-24</b>
Subject: <b>Computer Science</b>				
1.	Course Code			
2.	Course Title	<b>Programming using C++ and Data Structure</b>		
3.	Course Type (Core Course/Elective/Generic Elective/ Vocational	<b>Major/Minor</b>		
4.	Pre-Requisite (if any)	To study this course, a student must have basic knowledge of Computers.		
5.	Course Learning Outcomes(CLO)	<b>After the completion of this course, a successful student will be able to do the following:</b>  1. Develop simple algorithms and flow charts to solve a problem with programming using top down design principles. 2. Writing efficient and well-structured computer algorithms/programs. 3. Learn to formulate iterative solutions and array processing algorithms for problems. 4. Use recursive techniques, pointers and searching methods in programming. 5. Will be familiar with fundamental data structures, their implementation; become accustomed to the description of algorithms in both functional and procedural styles. 6. Have knowledge of complexity of basic operations like insert, delete, search on these data structures. 7. Possess ability to choose a data structure to suitably model any data used in computer applications. 8. Design programs using various data structures including hash tables, Binary and general search trees, heaps, graphs etc. 9. Assess efficiency tradeoffs among different data structure implementations. 10. Implement and know the applications of algorithms for searching and sorting.		
6.	Credit Value	<b>Theory – 4 Credits Practical – 2 Credits</b>		
7.	Total Marks	Max. Marks : 40+ <b>60</b>		Min. Passing Marks: <b>35</b>
<b>PART B: Content of the Course</b>				



No. of Lectures (in hours per week): <b>4 Hrs. per week</b>		
Total No. of Lectures: <b>60.</b>		
Module	Topics	No. of Lectures
I	<b>Basics of OOPs:</b> Features and Characteristics of OOPs, History of C++, Application of C++, Data Types, Operator in C++, C++ Stream Classes, Unformatted and Formatted I/O Operation, Managing Output with Manipulators, Scope Resolution Operator	12
II	<b>Functions In C++:</b> The Main Function, Function Prototyping, Call by Reference Call by Address, Call by Value, Return by Reference, Inline Function, Default Arguments, Constant Arguments, Function Overloading, <b>Classes &amp; Objects:</b> A Sample C++ Program with class, Defining Member Functions (Private & Public), Static Data Members, Static Member, Functions, Array of Objects, Object as Function Arguments, Friend Functions.	12
III	<b>Arrays:</b> Representation of single, two-dimensional arrays <b>Constructor &amp; Destructor:</b> Constructor, Constructors with Default Arguments, Parameterized Constructor, Copy Constructor, Multiple Constructors in a Class, Destructor. <b>Searching</b> (linear & binary) and <b>sorting</b> (bubble sort, selection sort & insertion sorting)	12
IV	<b>Inheritance:</b> Defining Derived Classes, Single Inheritance, Making a Private Member Inheritable, Multilevel Inheritance, Hierarchical Inheritance, Multiple Inheritance, Hybrid Inheritance, Virtual Base Classes, Abstract Classes, Operator Overloading. <b>Polymorphism:</b> Virtual functions. <b>Pointers, Exception Handling</b>	12
V	<b>Data Structure:</b> Basic concepts, Linear and Non-Linear data structures <b>Stacks:</b> Operations, Array and Linked Implementations, Applications- Infix to Postfix Conversion, Infix to Prefix Conversion, Postfix Expression Evaluation. <b>Queues:</b> Definition, Operations, Array and Linked Implementations. Circular Queue-Insertion and Deletion Operations, Dequeue (Double Ended Queue), Priority Queue- Implementation. <b>Linked Lists:</b> Singly Linked Lists, Operations, Circularly linked lists- Operations Doubly Linked Lists- Operations, Doubly Circular Linked List.	12

## PART C: Learning Resources

### Textbooks, Reference Books, Other Resources

#### Suggested Readings

##### Textbooks:

- J. R. Hanly and E. B. Koffman, “Problem Solving and Program Design in C”, Pearson, 2015
- E. Balguruswamy, "C++ ", TMH Publication ISBN 0-07-462038-X
- Herbert Schildt, "C++ The Complete Reference "TMH Publication ISBN 0-07-463880-7

##### Reference Books:

- R. Lafore, 'Object Oriented Programming C++'
- N. Dale and C. Weems, “Programming and problem solving with C++: brief edition”, Jones & Bartlett Learning.
- Adam Drozdek, "Data Structures and algorithm in C++", Third Edition, Cengage Learning.
- Sartaj Sahani, “Data Structures, Algorithms and Applications with C++”, McGraw Hill.
- Robert L. Kruse, "Data Structures and Program Design in C++", Pearson.
- D.S. Malik, “Data Structure using C++”, Second edition, Cengage Learning.
- M. A. Weiss, “Data structures and Algorithm Analysis in C”, 2nd edition, Pearson.
- Lipschutz, “Schaum’s outline series Data structures”, Tata McGraw-Hill

#### Suggestive digital platform web links

<https://www.youtube.com/watch?v=BCIS40yzssA>

<https://www.youtube.com/watch?v=vLnPwxZdW4Y&vl=en>

<https://www.youtube.com/watch?v=Umm1ZQ5ltZw>

#### Suggested equivalent online courses

S.No.	Online Course	Duration	Platform
1	Programming in C++ <a href="https://nptel.ac.in/courses/106/105/106105151/">https://nptel.ac.in/courses/106/105/106105151/</a>	8 weeks	NPTEL
2	Beginning C++ Programming - From Beginner to Beyond <a href="https://www.udemy.com/course/beginning-c-plus-plus-programming/">https://www.udemy.com/course/beginning-c-plus-plus-programming/</a>	Self paced	Udemy

## PART D: Assessment and Evaluation

**Internal Assessment :** Continuous Comprehensive Evaluation (CCE) : 40 Marks

Three test will be taken of which best of two marks will be considered

**External Assessment:** University Exam (UE) : 60 Marks

Time : **02.00 Hours**

Objective type Text I	20 Marks	Section (A) : Very short questions (1 from each unit)	1 x 5 = 5 Marks
Class Test II (Subjective)	20 Marks	Section (B) : 5 Short Questions (200 Words Each)	4 x 5 = 20 Marks
Class Test III (Subjective)	20 Marks		7 x 5 = 35 Marks
Total	<b>40 Marks</b>	Total	<b>60 Marks</b>
Any remarks/suggestions: <b>Focus of the course/teaching should be on developing ability of the student in analyzing a problem, building the logic and efficient code for the problem.</b>			

<b>PART A: Introduction</b>			
Program: <b>Certificate</b>		Class: <b>B.Sc.</b>	Year: <b>I (sem 2)</b>
Session: <b>2023-24</b>			
Subject: <b>Computer Science</b>			
1.	Course Code		
2.	Course Title	<b>Programming using C++ Lab</b>	
3.	Course Type (Core Course/Elective/Generic Elective/ Vocational)	<b>Core Course</b>	
4.	Pre-Requisite (if any)	To study this course, a student must have basic logical and analytical skills.	
5.	Course Learning Outcomes(CLO)	<b>After the completion of this course, a successful student will be able to do the following:</b> <ol style="list-style-type: none"> <li>1. Develop simple algorithms and flow charts to solve a problem with programming using top down design principles.</li> <li>2. Writing efficient and well-structured computer algorithms/programs.</li> <li>3. Learn to formulate iterative solutions and array processing algorithms for problems.</li> <li>4. Use recursive techniques, pointers and searching methods in programming.</li> <li>5. Possess ability to choose a data structure to suitably model any data used in computer applications.</li> <li>6. Implement and know the applications of algorithms for searching and sorting etc.</li> </ol>	
6.	Credit Value	<b>Practical – 2 Credits</b>	
7.	Total Marks	Max. Marks : 40+ <b>60</b>	Min. Passing Marks: <b>35</b>
<b>PART B: Content of the Course</b>			
No. of Lab Practicals (in hours per week): <b>2 hours per week</b>			
Total No. of Lab.: <b>15 (30 hrs)</b>			
	<b>Suggestive list of Practicals</b>		<b>No. of Labs.</b>
	<b>Given the problem statement, students are required to formulate problem, develop flowchart/algorithm, write code in C++, execute and test it. Students should be given assignments on following :</b>		15

	<ol style="list-style-type: none"> <li>1. Write a program to find area of a circle, rectangle, square using switch case.</li> <li>2. Write a program to convert decimal (integer) number into equivalent binary number.</li> <li>3. Write a program to check given string is palindrome or not.</li> <li>4. Write a program to print digits of entered number in reverse order.</li> <li>5. Write a program to print sum of two matrices.</li> <li>6. Write a program whether a given number is prime or not.</li> <li>7. Write a program to check entered number is Armstrong or not.</li> <li>8. Write a program to find the area and volume of a rectangular box using constructor.</li> <li>9. Write a program to implement single inheritance.</li> <li>10. Write a program to find largest element from an array.</li> <li>11. Write a program to implement push and pop operations on a stack using array.</li> <li>12. Write a program to perform insert and delete operations on a queue using array.</li> <li>13. Write a program for Linear search.</li> <li>14. Write a program for Binary search.</li> <li>15. Write a program for Bubble sort.</li> <li>16. Write a program for Selection sort.</li> <li>17. Write a program for Insertion sort.</li> <li>18. Write a program to implement linked list.</li> </ol>	
--	--	--

### PART C: Learning Resources

#### Textbooks, Reference Books, Other Resources

#### Suggested Readings

- J. R. Hanly and E. B. Koffman, "Problem Solving and Program Design in C", Pearson, 2015
- E. Balguruswamy, "C++ ", TMH Publication ISBN O-07-462038-X
- Herbert Schildt, "C++ The Complete Reference "TMH Publication ISBN 0-07-463880-7

#### Reference Books:

- R. Lafore, 'Object Oriented Programming C++'
- N. Dale and C. Weems, "Programming and problem solving with C++: brief edition", Jones & Bartlett Learning.
- Adam Drozdek, "Data Structures and algorithm in C++", Third Edition, Cengage Learning.
- Sartaj Sahani, "Data Structures, Algorithms and Applications with C++", McGraw Hill.
- Robert L. Kruse, "Data Structures and Program Design in C++", Pearson.
- D.S. Malik, "Data Structure using C++", Second edition, Cengage Learning.
- M. A. Weiss, "Data structures and Algorithm Analysis in C", 2nd edition, Pearson.
- Lipschutz, "Schaum's outline series Data structures", Tata McGraw-Hill

#### Suggestive digital platform web links

<https://www.youtube.com/watch?v=BCIS40yzssA>

<https://www.youtube.com/watch?v=vLnPwxZdW4Y&vl=en>

<https://www.youtube.com/watch?v=Umm1ZQ5ltZw>

Suggested equivalent online courses

S.No.	Online Course	Duration	Platform	
1	Programming in C++ <a href="https://nptel.ac.in/courses/106/105/106105151/">https://nptel.ac.in/courses/106/105/106105151/</a>	8 weeks	NPTEL	
2	Beginning C++ Programming - From Beginner to Beyond <a href="https://www.udemy.com/course/beginning-c-plus-plus-programming/">https://www.udemy.com/course/beginning-c-plus-plus-programming/</a>	Self paced	Udemy	

#### PART D: Assessment and Evaluation

**Internal Assessment :** Continuous Comprehensive Evaluation (CCE) : 40 Marks

**External Assessment:** University Exam (UE) : 60 Marks

Time : 02.00 Hours

Internal Assessment	Marks	External Assessment	Marks
Lab Attendance	10 Marks	Practical record file	25 Marks
		Viva voce practical	10 Marks
Internal Viva	10 Marks	Execution	05 Marks
Practical File	20 Marks	Answer script	20 Marks
<b>Total</b>	<b>40 Marks</b>	<b>Total</b>	<b>60 Marks</b>

# **ST. ALOYSIUS COLLEGE (AUTONOMOUS), JABALPUR**

## **PART A: Introduction**

Program: <b>Certificate</b>	Class: <b>BSc</b>	Year: <b>I (sem 2)</b>	Session: <b>2023-24</b>
Subject: <b>Computer Science</b>			
1.	Course Code		
2.	Course Title	<b>Programming using C++</b>	
3.	Course Type (Core Course/Elective/Generic Elective/ Vocational)	<b>Electives</b>	
4.	Pre-Requisite (if any)	To study this course, a student must have basic knowledge of Computers.	
5.	Course Learning Outcomes(CLO)	<b>After the completion of this course, a successful student will be able to do the following:</b>  1. Develop simple algorithms and flow charts to solve a problem with programming using top down design principles. 2. Writing efficient and well-structured computer algorithms/programs. 3. Learn to formulate iterative solutions and array processing algorithms for problems. 4. Use recursive techniques, pointers and searching methods in programming.	
6.	Credit Value	<b>Theory – 3 Credits Practical – 1 Credits</b>	
7.	Total Marks	Max. Marks : 40+60	Min. Passing Marks: <b>35</b>

## **PART B: Content of the Course**

No. of Lectures (in hours per week): **4 Hrs. per week**

Total No. of Lectures: **45.**

<b>Module</b>	<b>Topics</b>	<b>No. of Lectures</b>
I	<b>Basics of OOPs:</b> Features and Characteristics of OOPs, History of C++, Application of C++, Data Types, Operator in C++, C++ Stream Classes, Unformatted and Formatted I/O Operation, Managing Output with Manipulators, Scope Resolution Operator	15
II	<b>Functions In C++:</b> The Main Function, Function Prototyping, Call by Reference Call by Address, Call by Value, Return by Reference, Inline Function, Default Arguments, Constant Arguments, Function Overloading,	10

	<b>Classes &amp; Objects:</b> A Sample C++ Program with class, Defining Member Functions (Private & Public), Static Data Members, Static Member, Functions, Array of Objects, Object as Function Arguments, Friend Functions.	
III	<b>Arrays:</b> Representation of single, two-dimensional arrays <b>Constructor &amp; Destructor:</b> Constructor, Constructors with Default Arguments, Parameterized Constructor, Copy Constructor, Multiple Constructors in a Class, Destructor. <b>Searching</b> (linear & binary) and <b>sorting</b> (bubble sort, selection sort & insertion sorting)	10
IV	<b>Inheritance:</b> Defining Derived Classes, Single Inheritance, Making a Private Member Inheritable, Multilevel Inheritance, Hierarchical Inheritance, Multiple Inheritance, Hybrid Inheritance, Virtual Base Classes, Abstract Classes, Operator Overloading. <b>Polymorphism:</b> Virtual functions. <b>Pointers, Exception Handling</b>	10

### PART C: Learning Resources

Textbooks, Reference Books, Other Resources

Suggested Readings

#### Textbooks:

- J. R. Hanly and E. B. Koffman, "Problem Solving and Program Design in C", Pearson, 2015
- E. Balguruswamy, "C++ ", TMH Publication ISBN O-07-462038-X
- Herbert Schildt, "C++ The Complete Reference "TMH Publication ISBN 0-07-463880-7

#### Reference Books:

- R. Lafore, 'Object Oriented Programming C++'
- N. Dale and C. Weems, "Programming and problem solving with C++: brief edition", Jones & Bartlett Learning.

Suggestive digital platform web links

<https://www.youtube.com/watch?v=BCIS40yzssA>

<https://www.youtube.com/watch?v=vLnPwxZdW4Y&vl=en>

<https://www.youtube.com/watch?v=Umm1ZQ5ltZw>

Suggested equivalent online courses

S.No.	Online Course	Duration	Platform
1	Programming in C++	8 weeks	NPTEL



	<a href="https://nptel.ac.in/courses/106/105/106105151/">https://nptel.ac.in/courses/106/105/106105151/</a>		
2	Beginning C++ Programming - From Beginner to Beyond <a href="https://www.udemy.com/course/beginning-c-plus-plus-programming/">https://www.udemy.com/course/beginning-c-plus-plus-programming/</a>	Self paced	Udemy

**PART D: Assessment and Evaluation**

<b>Internal Assessment</b> : Continuous Comprehensive Evaluation (CCE) : 40 <b>Marks</b>  Three test will be taken of which best of two marks will be considered		<b>External Assessment:</b> University Exam (UE) : 60 <b>Marks</b>  Time : <b>02.00 Hours</b>	
Objective type Text I	20 Marks	Section (A) : Very short questions (1 from each unit)	1 x 5 = 5 Marks
Class Test II (Subjective)	20 Marks	Section (B) : 5 Short Questions (200 Words Each)	4 x 5 = 20 Marks
Class Test III (Subjective)	20 Marks		7 x 5 = 35 Marks
Total	<b>40 Marks</b>	Total	<b>60 Marks</b>

Any remarks/suggestions: **Focus of the course/teaching should be on developing ability of the student in analyzing a problem, building the logic and efficient code for the problem.**

<b>PART A: Introduction</b>			
Program: <b>Certificate</b>		Class: <b>B.Sc</b>	Year: <b>I (Sem 2)</b>
Session: <b>2023-24</b>			
Subject: <b>Computer Science</b>			
1.	Course Code		
2.	Course Title	<b>Programming using C++ Lab</b>	
3.	Course Type (Core Course/Elective/Generic Elective/ Vocational)	<b>Electives</b>	
4.	Pre-Requisite (if any)	To study this course, a student must have basic logical and analytical skills.	
5.	Course Learning Outcomes(CLO)	<b>After the completion of this course, a successful student will be able to do the following:</b> <ol style="list-style-type: none"> <li>1. Develop simple algorithms and flow charts to solve a problem with programming using top down design principles.</li> <li>2. Writing efficient and well-structured computer algorithms/programs.</li> <li>3. Learn to formulate iterative solutions and array processing algorithms for problems.</li> <li>4. Use recursive techniques, pointers and searching methods in programming.</li> <li>5. Implement and know the applications of algorithms for searching and sorting etc.</li> </ol>	
6.	Credit Value	<b>Practical – 1 Credits</b>	
7.	Total Marks	Max. Marks : 40+ <b>60</b>	Min. Passing Marks: <b>35</b>
<b>PART B: Content of the Course</b>			
No. of Lab Practical (in hours per week): <b>2 hours per week</b>			
Total No. of Lab.: <b>8 (16 hrs)</b>			
	<b>Suggestive list of Practical</b>		<b>No. of Labs.</b>
	<b>Given the problem statement, students are required to formulate problem, develop flowchart/algorithm, write code in C++, execute and test it. Students should be given assignments on following :</b> <ol style="list-style-type: none"> <li>1. Write a program to find area of a circle, rectangle, square using switch case.</li> </ol>		8

	<ol style="list-style-type: none"> <li>2. Write a program to convert decimal (integer) number into equivalent binary number.</li> <li>3. Write a program to check given string is palindrome or not.</li> <li>4. Write a program to print digits of entered number in reverse order.</li> <li>5. Write a program to print sum of two matrices.</li> <li>6. Write a program whether a given number is prime or not.</li> <li>7. Write a program to check entered number is Armstrong or not.</li> <li>8. Write a program to find the area and volume of a rectangular box using constructor.</li> <li>9. Write a program for Linear search.</li> <li>10. Write a program for Binary search.</li> <li>11. Write a program for Bubble sort.</li> <li>12. Write a program for Selection sort.</li> <li>13. Write a program for Insertion sort.</li> </ol>	
--	---	--

### PART C: Learning Resources

#### Textbooks, Reference Books, Other Resources

#### Suggested Readings

- J. R. Hanly and E. B. Koffman, "Problem Solving and Program Design in C", Pearson, 2015
- E. Balguruswamy, "C++ ", TMH Publication ISBN O-07-462038-X
- Herbert Schildt, "C++ The Complete Reference "TMH Publication ISBN 0-07-463880-7

#### Reference Books:

- R. Lafore, 'Object Oriented Programming C++'
- N. Dale and C. Weems, "Programming and problem solving with C++: brief edition", Jones & Bartlett Learning.

#### Suggestive digital platform web links

<https://www.youtube.com/watch?v=BCIS40yzssA>

<https://www.youtube.com/watch?v=vLnPwxZdW4Y&vl=en>

<https://www.youtube.com/watch?v=Umm1ZQ5ltZw>

#### Suggested equivalent online courses

S.No.	Online Course	Duration	Platform	
1	Programming in C++ <a href="https://nptel.ac.in/courses/106/105/106105151/">https://nptel.ac.in/courses/106/105/106105151/</a>	8 weeks	NPTEL	
2	Beginning C++ Programming - From Beginner to Beyond <a href="https://www.udemy.com/course/beginning-c-plus-plus-programming/">https://www.udemy.com/course/beginning-c-plus-plus-programming/</a>	Self paced	Udemy	

**PART D: Assessment and Evaluation**

<b>Internal Assessment</b> : Continuous Comprehensive Evaluation (CCE) : 40 <b>Marks</b>		<b>External Assessment:</b> University Exam (UE) : <b>60 Marks</b>  Time : <b>02.00 Hours</b>	
<b>Internal Assessment</b>	<b>Marks</b>	<b>External Assessment</b>	<b>Marks</b>
Lab Attendance	10 Marks	Practical record file	25 Marks
		Viva voce practical	05 Marks
Internal Viva	10 Marks	Execution	10 Marks
Practical File	20 Marks	Answer script	20 Marks
<b>Total</b>	<b>40 Marks</b>	<b>Total</b>	<b>60 Marks</b>