

Government Polytechnic, Balangir							
Department of Computer Science Engineering							
LESSON PLAN 2026-27(WINTER)							
NAME OF THE TEACHER : DEEPAK KUMAR BARDHA, LECT.CSE(STAGE-II)							
Subject: PROGRAMMING WITH C++ (Course Code: CSEPC 201/TH1)							
Program: Diploma in Computer Science and Engineering							
Semester: 3rd							
Total Contact Hours: 45							
Total Marks: 100							
Assessment: Internal Assessment – 30, End Term – 70							
After completion of the course, the students will be able to:							
CO1: Describe object-oriented programming (OOP) principles.							
CO2: Develop proficiency in C++ syntax and programming constructs.							
CO3: Implement advanced OOP features for software design.							
CO4: Demonstrate polymorphism and operator overloading.							
CO5: Handle exceptions and ensure robust program execution.							
Period	Unit	Topic / Sub-Topic	Learning Objectives	Activity	Course Outcome	Learning Methodology	Homework
1	I	Introduction to object-oriented programming, encapsulation, polymorphism.	Understand core OOP concepts.	Interactive Q&A	CO1	Lecture & Discussion	Read textbook intro on OOP.
2	I	Getting started with C++ syntax, namespaces, data-type, variables.	Write a basic C++ program.	Coding Demo	CO2	Demonstration	Write a "Hello World" program.
3	I	User defined types, structures, unions.	Differentiate and use structures and unions.	Code Tracing	CO2	Lecture & Coding	Define a student structure.
4	I	Operators, flow control.	Apply operators and conditional statements.	Dry Run Exercises	CO2	Problem Solving	Practice conditional exercises.
5	I	Flow control (Loops).	Implement iterative execution.	Coding Demo	CO2	Demonstration	Write a loop to print a series.
6	I	Arrays.	Store and manipulate array data.	Array tracing	CO2	Lecture & Coding	Array sorting program.
7	I	Strings.	Perform string manipulation.	Code implementation	CO2	Demonstration	Program to reverse a string.
8	I	Pointers.	Understand memory addresses.	Memory mapping on board	CO2	Lecture & Visuals	Pointer declaration practice.
9	I	Pointers with arrays and strings.	Traverse arrays via pointers.	Code execution	CO2	Demonstration	Access array via pointers.
10	I	Functions, default values in functions.	Create reusable code blocks.	Coding Demo	CO2	Lecture & Coding	Write a function with default args.
11	I	Recursion.	Understand recursive calls.	Call-stack tracing	CO2	Problem Solving	Write a factorial program.
12	I	Unit I Review & Practice.	Consolidate Unit I concepts.	Quiz / Q&A	CO1, CO2	Discussion	Review Unit I notes.

13	II	Abstraction mechanism: Classes, private, public, member data, member functions.	Define a class and create objects.	Class design exercise	CO3	Lecture & Demo	Create a basic class program.
14	II	Inline function, references.	Optimize function calls.	Code comparison	CO3	Demonstration	Implement an inline function.
15	II	Friend functions, static members.	Share access across classes.	Coding Demo	CO3	Lecture & Coding	Write a friend function.
16	II	Constructors.	Initialize object data.	Concept mapping	CO3	Lecture	Write default/parameterized constructors.
17	II	Destructors.	Manage object cleanup.	Code tracing	CO3	Demonstration	Implement a destructor.
18	II	Inheritance: Class hierarchy, derived classes.	Understand code reusability.	Draw class hierarchies	CO3	Lecture & Visuals	Identify real-world hierarchies.
19	II	Single and multilevel inheritance.	Implement basic inheritance.	Coding Demo	CO3	Demonstration	Code a single inheritance model.
20	II	Multiple, hybrid inheritance.	Implement complex inheritance.	Code implementation	CO3	Lecture & Coding	Code a multiple inheritance model.
21	II	Role of virtual base class.	Solve the diamond problem.	Diamond problem tracing	CO3	Problem Solving	Practice virtual base class syntax.
22	II	Constructor and destructor execution.	Trace execution order.	Output prediction	CO3	Discussion	Predict outputs for constructor chains.
23	II	Base initialization using derived class constructors.	Pass values to base classes.	Coding Demo	CO3	Demonstration	Write a program using base initialization.
24	III	Polymorphism: Binding, Static binding, Dynamic binding.	Differentiate binding types.	Concept comparison	CO4	Lecture	Read about early vs late binding.
25	III	Static polymorphism: Function Overloading.	Overload standard functions.	Coding Demo	CO4	Demonstration	Overload an area() function.
26	III	Ambiguity in function overloading.	Identify and resolve errors.	Error debugging	CO4	Problem Solving	Fix an ambiguous code snippet.
27	III	Dynamic polymorphism: Base class pointer.	Point to derived objects.	Memory visual drawing	CO4	Lecture & Demo	Create a base class pointer.
28	III	Object slicing, late binding.	Prevent data loss in assignment.	Output prediction	CO4	Discussion	Write notes on object slicing.
29	III	Method overriding with virtual functions.	Override base methods safely.	Coding Demo	CO4	Lecture & Coding	Implement a virtual function.
30	III	Pure virtual functions.	Force derived class implementation	Code execution	CO4	Demonstration	Write a pure virtual function.

LESSON PLAN 3rd sem PROGRAMMING WITH C++

31	III	Abstract classes.	Create interface-like structures.	Architecture exercise	CO4	Lecture & Demo	Design an abstract Shape class.
32	III	Unit III Review & Practice.	Consolidate polymorphism.	Peer programming	CO4	Problem Solving	Complete textbook exercises.
33	IV	Operator Overloading: This pointer.	Understand current object context.	Code tracing	CO4	Lecture	Read about the 'this' pointer.
34	IV	Applications of this pointer.	Return object by reference.	Coding Demo	CO4	Demonstration	Chain function calls using 'this'.
35	IV	Operator function.	Understand operator syntax.	Syntax review	CO4	Lecture	Review operator precedence.
36	IV	Member operator function.	Overload operators inside class.	Coding Demo	CO4	Demonstration	Overload the '+' operator.
37	IV	Non-member operator function.	Overload using friend functions.	Code implementation	CO4	Lecture & Coding	Overload with a non-member function.
38	IV	Operator overloading (Unary).	Overload ++ and --.	Code execution	CO4	Demonstration	Write a program for ++ overloading.
39	IV	Operator overloading (Binary).	Overload +, -, *, etc.	Coding Demo	CO4	Problem Solving	Write a program for adding complex numbers.
40	IV	I/O operators.	Overload << and >>.	Code implementation	CO4	Demonstration	Overload standard I/O for a class.
41	IV	Unit IV Review & Practice.	Consolidate operator overloading.	Quiz / Code Review	CO4	Discussion	Complete practice assignment.
42	V	Exception handling: Try, throw, and catch.	Catch runtime errors.	Error simulation	CO5	Lecture & Demo	Write a divide-by-zero exception program.
43	V	Exceptions and derived classes.	Handle complex exceptions.	Coding Demo	CO5	Demonstration	Catch exceptions based on hierarchy.
44	V	Function exception declaration.	Restrict function exceptions.	Code execution	CO5	Lecture & Coding	Declare specific throws in functions.
45	V	Unexpected exceptions & Course Wrap-up.	Manage unhandled errors.	Q&A session	CO1-CO5	Discussion	Prepare for the final exam.

Deepak kumar Bando
25.06.2026
Signature of Teacher

Deepak kumar Bando
25.06.2026
Signature of HOD