Vidya Pratishthan's
Kamalnayan Bajaj Institute of
Engineering and Technology, Baramati.
(An Autonomous Institute)

Faculty of Science and Technology

Board of Studies

Computer Engineering

Syllabus

Multidisciplinary Minor

(Pattern 2024)
(w.e.f. AY: 2025-26)

Syllabus: Multidisciplinary Minor Computer Engineering

Pattern 2024 w.e.f. AY:2025-2026

SEMESTER-III,IV,V,VI,VII

Course Code	Courses Name	Teaching Scheme			Examination Scheme and Marks							Credits			
		TH	PR	TUT	CAA	ISE	ESE	TW	PR	OR	Total	TH	PR	TUT	Total
CO24051	Computer Graphics and Gaming	2	2	-	10	-	60	30	-	-	100	2	1		3
CO24052	Object Oriented Programming	3	2	-	10	30	60	30	-	-	130	3	1		4

TH: Theory PR : Practical TUT : Tutorial CAA : Continuous Activity Assessment , ISE : In Semester Examination , ESE : End Semester Examination TW : Term-Work , OR : Oral

Dept. Autonomy Coordinator
Mr. M. D. Shelar

Academic Coordinator
Dr. P. M. Paithane

Head of Department
Dr. G. J. Chhajed

Dean Autonomy
Dr. C. B. Nayak

Dean Academic
Dr. S. M. Bhosle

Principal
Dr. S. B. Lande

Principal
Vidya Pratishthan's
Kamalnayan Bajaj Institute of
Engineering & Technology, Baramati
Vidyanagari, Baramati-413133

VPKBIET BARAMATI
Affiliated to
S.P.P.U. Pune
Id No. /PU/PN/
Engg./152/2000
PUNE (MS.)

CO24051: Computer Graphics and Gaming

Teaching Scheme:	Credit: 03	Examination Scheme:
TH:02Hrs/Week PR:02Hrs/Week	TH Credit :02 PR Credit :01	Course Activity :20Marks In Semester :20Marks End Semester :50Marks Term-Work :20Marks

Prerequisite:

C++ programming (CPP)

Companion Course, if any: Computer Graphics Laboratory

Course Objective:

1. Remembering: To acquaint the learner with the basic concepts of Computer Graphics
2. Understanding: To learn the various algorithms for generating and rendering graphical figures.
3. Applying: To get familiar with mathematics behind the graphical transformations
4. Understanding: To understand and apply various methods and techniques regarding animation.
5. Creating: To generate Interactive graphics using OpenGL

Course Outcomes:

Students will be able to

1. Be familiar with the graphics designing concepts and devices.
2. Construct a mathematical design using the development process.
3. Recognize the design principles of animation and gaming application.
4. Implement the use of gaming tools in application design.

Course Activity :

The course coordinator should identify relative and innovative activities for course activity. Below are some suggested course activity for course coordinator

1. Active participation in Gaming Competition
2. Poster Presentation
3. Video Presentations
4. Survey on various Animation making tools
5. Visit to Animation Business Schools

Course Contents

Mapping of Course Outcomes for Unit I	CO1

UNIT I	Basics of Computer Graphics	07 Hours

Introduction, What is computer Graphics? Area of Computer Graphics, Design and Drawing, Animation Multimedia applications, Simulation, How are pictures actually stored and displayed, Difficulties for displaying pictures. Graphics Devices Cathode Ray Tube, Quality of Phosphors, CRTs for Color Display, Beam Penetration CRT, The Shadow - Mask CRT, Direct View Storage Tube, Tablets, The light Pen, Three Dimensional Devices.

Mapping of Course Outcomes for Unit II	CO2

UNIT II	Two and Three Dimensional Transformations	07 Hours

Simple line drawing methods, Introduction Point Plotting Techniques Qualities of good line drawing algorithms The Digital Differential Analyzer (DDA), Bresenham's Algorithm Generation of Circles Introduction, what is transformation? Matrix representation of points Basic transformation, Translation, Rotation, Scaling, Need for 3-Dimensional Imaging Techniques for 3-Dimesional displaying, Translation, Rotation, Scaling

Mapping of Course Outcomes for Unit III	CO3

UNIT III	Animation	07 Hours

Animation: Introduction, Conventional and computer-based animation, Segment: Introduction, Segment table, Segment creation, closing, deleting and renaming, Visibility. Design of animation sequences, Animation languages, Key- frame, Morphing, Motion specification. Gaming: Introduction, Gaming platform (NVIDIA, i8060), Advances in Gaming.

Mapping of Course Outcomes for Unit IV	CO4

UNIT IV	Gaming	07Hours

Principles of game design, Game Design Theory,MDA,8 type of Fun in Game, Visual style, Gameplay, Generate ideas for a game concept Idea Development Process, Stimulus, Genre Market Research, Target platform ,Creating Prototype Creating physical Games: Board Game, Card Game, Party Games and etc….

Books and Other Resources

TextBooks:

1. Computer Graphics, Multimedia and Animation ,2010, Pakhira Malay K.
2. Donald D. Hearn and Baker- Computer Graphics with OpenGL, 4th Edition, ISBN-13: 9780136053583

Reference Books:

3. J. Foley, V. Dam, S. Feiner, J. Hughes, ―Computer Graphics Principles and Practicel, 2nd Edition,Pearson Education, 2003, ISBN 81 – 7808 – 038 – 9.
4. D. Rogers, J. Adams, ―Mathematical Elements for Computer Graphicsl, 2nd Edition, Tata McGrawHill Publication, 2002, ISBN 0 – 07 – 048677 – 8.

Guidelines for Term Work Assessment :

Term work assessment will be based on overall performance of Laboratory assignments performed by a students.

Guideline for Practical Conduction :

Use of open source software is encouraged. Based on the concepts learned. Instructor may also set one assignment or mini-project that is suitable to respective branch beyond the scope of syllabus.
Operating System recommended :- 64-bit Open source Linux or its derivative, Windows
Programming tools recommended: - Open Source C++ Programming tool like G++/GCC, OPENGL, DEV C++.

Guidelines for Practical Examination :

Problem statements will be formed based on assignments and performance will be evaluated by Internal and External Examiner. Relevant questions may be asked at the time of evaluation to test the student‗s understanding of the fundamentals, effective and efficient implementation.

Practical Assignments

1.	Write C++ program to draw the line styles using DDA and Bresenham‗s algorithm
2.	Write C++ program to draw a 4X4 chessboard.
3.	Write C++ program to draw 2-D object and perform following basic transformations, a) Scaling b) Translation c) Rotation.
4.	Write C++ program to draw Man Walking in the Rain with an Umbrella.
5.	Write a C++ Program to make puzzle game.
6.	Write a C++ Program to make Tic Tac Toe game.
7.	Write a C++ Program to draw a car in motion.

CO24053 : Object Oriented Programming

Teaching Scheme		Credit: 04	Examination Scheme	
TH	04 Hrs/Week		CAA:	10 Mark
			In Semester:	30 Mark
			End Semester:	60 Mark
PR	02 Hrs/Week	TH Credit :03 PR Credit :01	Term work:	25 Mark

Prerequisite: Fundamentals of Programming and Problem Solving

Course Objective:

- To learn the fundamental properties of the C++ language and its application.
- To understand the concept of function and polymorphism in C++.
- To learn design of classes for code reuse and implementation of dynamic binding using polymorphism
- To learn exception handling mechanism and various streams available for handling input and output and file handling mechanism
- To learn designing and implementation of generic programming using concept of template.
- To understand the basics of Data Structures

Course Outcomes:

CO1: **Design** the solution to a problem using object oriented programming concepts

CO2: **Reuse** the code using inheritance

CO3: **implementation** of run time and compile time polymorphism

CO4: **Differentiate** between various STL containers and select appropriate based on problem.

CO5: **Implement** file handling operations like open, close, read and write to files.

CO6: **Study** of basic data structures and its implementation

Course Activity (Any one):

- Simple Mini-Project in a group of 4 students for using C++ features and files on any topic
    - Banking Application
    - Students Results System
    - Library Management
    - Store Management
    - Billing System

Course Contents

Mapping of Course Outcomes for Unit I		CO1
UNIT I	**Fundamentals of Object Oriented Programming**	**08 Hours**

Introduction to OOP:
Need of object-oriented programming, Features of OOP: Namespaces, objects, classes, data members, methods, messages, data encapsulation, data abstraction and information hiding, inheritance, polymorphism.

C++ Programming- Data Types, Structures, Enumerations, control structures, Arrays and Strings, Class, Object, class and data abstraction, Access specifiers, separating interface from implementation.

Functions- Function, function prototype, accessing function and utility function, Constructors and destructor, Types of constructor, Objects and Memory requirements, Static members: variable and functions, inline function, friend function.

Mapping of Course Outcomes for Unit II		CO2
UNIT II	**Inheritance and Pointers**	**06 Hours**

Inheritance- Base Class and derived Class, protected members, relationship between base Class and Derived Class, Constructor and destructor in Derived Class, Overriding Member Functions, Class Hierarchies, Public and Private Inheritance, Types of Inheritance, Ambiguity in Multiple Inheritance, Virtual Base Class, Abstract class, Friend Class, Nested Class.

Pointers: declaring and initializing pointers, indirection Operators, Memory Management: new and delete, Pointers to Objects, this pointer, Pointers vs Arrays, accessing Arrays using pointers, Arrays of Pointers, Function pointers, Null pointer, void pointer.

Mapping of Course Outcomes for Unit III	**CO3**

UNIT III	**Polymorphism**	**06 Hours**

Polymorphism- Introduction to Polymorphism, Types of Polymorphism, Operator Overloading- concept of overloading, operator overloading, Overloading Unary Operators, Overloading Binary Operators, Data Conversion, Type casting (implicit and explicit), Pitfalls of Operator Overloading and Conversion, Keywords explicit and mutable. **Function overloading, Run Time Polymorphism**- Pointers to Base class, virtual function and its significance in C++, pure virtual function, abstract base class

Mapping of Course Outcomes for Unit IV	**CO4**

UNIT IV	**Standard Template Library**	**08 Hours**

Templates- The Power of Templates, Function template, class templates, overloading Function templates. template and friends Generic Functions, The type name and export keywords

Introduction to STL, STL Components, Containers- Sequence container and associative containers. container adapters, Application of Container classes: vector, list,

Algorithms- basic searching and sorting algorithms, min-max algorithm, set operations, heap sort

Iterators- input, output, forward, bidirectional and random access. Object Oriented Programming a road map to future

Mapping of Course Outcomes for Unit V	**CO5**

UNIT V	**Exception Handling and Files**	**06 Hours**

Exception Handling- Fundamentals, other error handling techniques, simple exception handling- Divide by Zero, Multiple catching, re-throwing an exception, exception specifications, user defined exceptions, processing unexpected exceptions

Files: Stream and files, Stream Classes, Stream Errors, File Pointers, File Operations : Open, Close, Read. Write

Mapping of Course Outcomes for Unit VI	**CO6**

UNIT VI	**Introduction Data Structures using C++**	**06 Hours**

Introduction to Data Structures, Need of Data Structures, Advantages of Data Structures, Stack, Queue, Searching, Sorting, Linked List(Linear and Single), ***Implementation must be known to students**

Books and Other Resources

Text Book:

1. Balagurusamy E, Object Oriented Programming with C++, Tata McGraw Hill Education Pvt.Ltd , Fourth Edition 2010
2. Sahay S. Object oriented Programming with C++. Oxford University Press; 2012.

Reference Books:

1. Daya Sagar Baral and Diwakar Baral, "The Secrets of Object Oriented Programming in C++", 1 st Edition 2010, Bhundipuran Prakasan
2. Lafore R. Object oriented programming in Turbo C++. Galgotia publications; 2001.
3. D. S. Malik, "C++ Programming", 3rd Edition 2007, Thomson Course Technology

4. Herbert Schildt, "C++: The Complete Reference", 4th Edition 2003, Tata McGraw Hill

Guidelines for Term Work Assessment : Term-work will be awarded on following parameters

- Timely submission of tutorial assignments
- Assessment marks awarded to each tutorial
- Theory and Tutorial attendance
- Performance in internal tests and in-semester examination

Practical Assignments

1. Write a C++ program to perform all arithmetic operations on simple integers
2. Write a C++ Program to calculate the percentage marks of a student (Assume End-Semester examination)
3. Write a C++ Program to create the database of student and perform operations like display information, add new information, modify information and delete the information. *Use Constructor, destructor, new, delete oprators
4. Write a C++ Program to create database of car of different brands. Create a base class for CAR and then derived class for its models. * Use inheritance and assume suitable data.
5. Write a C++ program for complex number operations using operator overloading concept
6. Write a C++ Program to function template to sort the integers, float and characters
7. Write a C++ program using STL using map associative container for state and its population. State name will be the key.
8. Write a C++ Program to create database of students and academic information using FILES.
9. Write a C++ program to implement the concept of Linked list. Operations expected are create link list, display link list, add new element in linked list and delete element from linked list.

Object Oriented Programming

Teaching Scheme		Credit: 04	Examination Scheme	
TH	04 Hrs/Week		Course Activity:	10 Mark
			In Semester:	30 Mark
			End Semester:	60 Mark
PR	02 Hrs/Week	TH Credit :03 PR Credit :01	Term work:	25 Mark

Prerequisite: Fundamentals of Programming and Problem Solving

Course Objective:

- To learn the fundamental properties of the C++ language and its application.
- To understand the concept of function and polymorphism in C++.
- To learn design of classes for code reuse and implementation of dynamic binding using polymorphism
- To learn exception handling mechanism and various streams available for handling input and output and file handling mechanism
- To learn designing and implementation of generic programming using concept of template.
- To understand the basics of Data Structures

Course Outcomes:

CO1: **Design** the solution to a problem using object oriented programming concepts

CO2: **Reuse** the code using inheritance

CO3: **implementation** of run time and compile time polymorphism

CO4**: Implement** file handling operations like open, close, read and write to files.

CO5: **Apply** exception handling to process the known errors while programming.

CO6: **Differentiate** between various STL containers and select appropriate based on problem.

Course Activity (Any one):

- *Simple Mini-Project in a group of 4 students for using C++ features and files on any topic*
  - o *Banking Application*
  - o *Students Results System*
  - o *Library Management*
  - o *Store Management*
  - o *Billing System*

Course Contents

Mapping of Course Outcomes for Unit I	CO1	
UNIT I	**Fundamentals of Object Oriented Programming**	**08 Hours**

Introduction to OOP:

Need of object-oriented programming, Features of OOP: Namespaces, objects, classes, data members, methods, messages, data encapsulation, data abstraction and information hiding, inheritance, polymorphism.

C++ Programming- Data Types, Structures, Enumerations, control structures, Arrays and Strings, Class, Object, class and data abstraction, Access specifiers, separating interface from implementation.

Functions- Function, function prototype, accessing function and utility function, Constructors and destructor, Types of constructor, Objects and Memory requirements, Static members: variable and functions, inline function, friend function.		
Mapping of Course Outcomes for Unit II	CO2	
UNIT II	**Inheritance and Pointers**	**06 Hours**
Inheritance- Base Class and derived Class, protected members, relationship between base Class and Derived Class, Constructor and destructor in Derived Class, Overriding Member Functions, Class Hierarchies, Public and Private Inheritance, Types of Inheritance, Ambiguity in Multiple Inheritance, Virtual Base Class, Abstract class, Friend Class, Nested Class. **Pointers**: declaring and initializing pointers, indirection Operators, Memory Management: new and delete, Pointers to Objects, this pointer, Pointers vs Arrays, accessing Arrays using pointers, Arrays of Pointers, Function pointers, Null pointer, void pointer.		
Mapping of Course Outcomes for Unit III	**CO3**	
UNIT III	**Polymorphism**	**06 Hours**
Polymorphism- Introduction to Polymorphism, Types of Polymorphism, Operator Overloading- concept of overloading, operator overloading, Overloading Unary Operators, Overloading Binary Operators, Data Conversion, Type casting (implicit and explicit), Pitfalls of Operator Overloading and Conversion, Keywords explicit and mutable. **Function overloading, Run Time Polymorphism-** Pointers to Base class, virtual function and its significance in C++, pure virtual function, abstract base class		
Mapping of Course Outcomes for Unit IV	**CO4**	
UNIT IV	**File Handling in C++**	**06 Hours**
Introduction to File Handling: Definition and need, C++ file libraries (fstream, ifstream, ofstream), Types of files: Text files vs Binary files, File stream classes: ifstream, ofstream, fstream, Opening and closing files and its modes, Opening and Closing Files using constructors Writing to Files, Reading from file, File Position Pointers, Error Handling in File Operations		
Mapping of Course Outcomes for Unit V	**CO5**	
UNIT V	**Exception Handling and Templates**	**06 Hours**
Exception Handling- Fundamentals, other error handling techniques, simple exception handling- Divide by Zero, Multiple catching, re-throwing an exception, exception specifications, user defined exceptions, processing unexpected exceptions **Templates-** The Power of Templates, Function template, class templates, overloading Function templates, template and friends Generic Functions, The type name and export keywords		
Mapping of Course Outcomes for Unit VI	**CO6**	
UNIT VI	**Standard Template Library**	**06 Hours**
Introduction to STL, STL Components, Containers- Sequence container and associative containers, container adapters, Application of Container classes: vector, list, **Algorithms-** basic searching and sorting algorithms, min-max algorithm, set operations, heap sort **Iterators-** input, output, forward, bidirectional and random access. Object Oriented Programming a road map to future		

Books and Other Resources

Text Book:

1. Object Oriented Programming with C++ , 8th Edition, E.Balagurusamy 2020
2. C++ Programming: An Object-Oriented Approach, First Edition, Tata McGraw Hill 2022

Reference Books:

1. Herbert Schildt, "C++: The Complete Reference", 4th Edition 2003, Tata McGraw Hill
2. Lafore R. Object oriented programming in C++. 4th Edition Pearson Eduction
3. D. S. Malik, "C++ Programming", 8th Edition 2017, Cengage Learning

Online References:

1. https://www.programiz.com/cpp-programming
2. https://www.geeksforgeeks.org/c-plus-plus/

NPTEL Course

- Programming in Modern C++
  - https://onlinecourses.nptel.ac.in/noc24_cs44/preview

Guidelines for Term Work Assessment : Term-work will be awarded on following parameters

- Timely submission of tutorial assignments
- Assessment marks awarded to each tutorial
- Theory and Tutorial attendance
- Performance in internal tests and in-semester examination

Practical Assignments

1. Write a program to perform all arithmetic operations on simple integers
2. Write a Program to calculate the percentage marks of a student (Assume End-Semester examination)
3. Write a Program to create the database of student and perform operations like display information, add new information, modify information and delete the information. *Use Constructor, destructor, new, delete operators
4. Write a Program to create database of car of different brands. Create a base class for CAR and then derived class for its models. * Use inheritance and assume suitable data.
5. Write a program for complex number operations using operator overloading concept
6. Write a Program to function template to sort the integers, float and characters
7. Write a program using STL using map associative container for state and its population. State name will be the key.
8. Write a Program to create database of students and academic information using FILES.