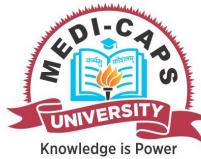


*Total No. of Questions: 6*

*Total No. of Printed Pages:3*

**Enrollment No.....**





**Faculty of Engineering / Science**  
**End Sem Examination Dec 2024**

**CS3CO30 / BC3CO35 Object Oriented Programming**

Programme: B.Tech./B.Sc. Branch/Specialisation: CSE All / Computer Science

**Duration: 3 Hrs.** **Maximum Marks: 60**

Duration: 3 Hrs.

Note: All questions are compulsory. Internal choices, if any, are indicated. Answers of Q.1 (MCQs) should be written in full instead of only a, b, c or d. Assume suitable data if necessary. Notations and symbols have their usual meaning.

	[2]		[3]
vi.	What is the main advantage of recursive association in OOP? (a) Simplicity (b) Reusability (c) Modeling self-referential objects (d) Memory optimization	1 1 1 03 2	
vii.	Which type of polymorphism is achieved through method overloading? (a) Static (b) Dynamic (c) Runtime (d) None of these	1 1 1 04 3	
viii.	What is the main purpose of virtual functions in OOP? (a) Compile-time binding (b) Dynamic binding (c) Abstract classes (d) Constructor chaining	1 1 1 04 3	
ix.	Which container type allows heterogeneous object storage? (a) Array (b) Vector (c) List (d) Map	1 1 2 05 3	
x.	In OOP, streams are primarily used for: (a) Memory management (b) File I/O operations (c) Object instantiation (d) Code execution	1 1 2 05 3	
Q.2	i. Define object-oriented programming. What are its key characteristics? ii. Explain the concepts of abstraction, encapsulation and information hiding with examples of each.	4 1 1 01 1	
OR	iii. Compare object-oriented programming and procedural programming in terms of abstraction, modularity, reusability, and scalability. Give examples of each.	6 2 2 01 1	
			Q.3 i. Explain the concept of Meta-class and its role in object instantiation. ii. Explain the concept of object lifetime. How does encapsulation ensure data integrity during an object's lifecycle? Provide detailed examples.
			OR iii. Differentiate between static objects and dynamic objects in object-oriented programming. Explain their memory allocation and lifetime.
			Q.4 i. Define aggregation in OOP. Explain how it differs from association. ii. Discuss multiplicities and navigability in associations. How are they modeled in OOP?
			OR iii. Explain the concept of association between objects. Differentiate between unidirectional and bidirectional associations with suitable examples.
			Q.5 i. What is dynamic polymorphism? Explain with examples. ii. Explain the concept of multiple inheritance with a suitable example. Highlight the advantages and challenges of multiple inheritance in OOP.
			OR iii. Explain the difference between public and protected access specifiers in C++. How does inheritance affect their visibility? Give examples.
			Q.6 Attempt any two: i. Discuss the concept of persistent objects in OOP. How are they managed using streams and files? ii. Explain the purpose and usage of container classes in object-oriented programming. Provide an example to demonstrate their importance. iii. What are heterogeneous containers? Explain their purpose and demonstrate with an example.

\*\*\*\*\*

**Marking Scheme**  
**Object Oriented Programming CS3CO30-BC3CO35**

Q.1	i) a) Objects	1		OR	iii. Differentiate between static objects and dynamic objects in Object-Oriented Programming. <b>3 Marks</b> Explain their memory allocation and lifetime. <b>3 Marks</b>	6
	ii) b) Encapsulation	1		Q.4	i. Define Aggregation in OOP. <b>2 Marks</b> Explain how it differs from Association. <b>2 Marks</b>	4
	iii) d) Dynamic	1		ii.	Discuss multiplicities and navigability in associations. <b>3 Marks</b> How are they modeled in OOP? <b>3 Marks</b>	6
	iv) b) Define object properties	1		OR	iii. Explain the concept of association between objects. <b>2 Marks</b> Differentiate between unidirectional and bidirectional associations <b>2 Marks</b> with suitable examples. <b>2 Marks</b>	6
	v) d) Aggregation	1		Q.5	i. What is dynamic polymorphism? <b>2 Marks</b> Explain with examples. <b>2 Marks</b>	4
	vi) c) Modeling self-referential objects	1		ii.	Explain the concept of multiple inheritance with a suitable example. <b>3 Marks</b> Highlight the advantages and challenges of multiple inheritance in OOP. <b>3 Marks</b>	6
	vii) a) Static	1		OR	iii. Explain the difference between public and protected access specifiers in C++. <b>2 Marks</b> How does inheritance affect their visibility? <b>2 Marks</b> Give examples. <b>2 Marks</b>	6
	viii) b) Dynamic binding	1		Q.6	i. Discuss the concept of persistent objects in OOP. <b>2.5 Marks</b> How are they managed using streams and files? <b>2.5 Marks</b>	5
	ix) d) Map	1		ii.	Explain the purpose and usage of container classes in Object-Oriented Programming. <b>2.5 Marks</b> Provide an example to demonstrate their importance. <b>2.5 Marks</b>	5
	x) b) File I/O operations	1		iii.	What are heterogeneous containers? <b>2.5 Marks</b> Explain their purpose and demonstrate with an example. <b>2.5 Marks</b>	5
Q.2	i. Define Object-Oriented Programming. <b>2 Marks</b> What are its key characteristics? <b>2 Marks</b>	4				
	ii. Explain the concepts of Abstraction, Encapsulation and Information Hiding with examples of each. <b>2 Marks Each</b>	6				
	iii. Compare Object-Oriented Programming and Procedural Programming in terms of abstraction, modularity, reusability, and scalability. Give examples of each. <b>1.5 marks Each topic</b>	6				
	iv.					
Q.3	i. Explain the concept of Meta-class <b>2 Marks</b> and its role in object instantiation. <b>2 Marks</b>	4				
	ii. Explain the concept of object lifetime. <b>2 Marks</b> How does encapsulation ensure data integrity during an object's lifecycle? <b>2 Marks</b> Provide detailed examples. <b>2 Marks</b>	6				

[2]

\*\*\*\*\*

[3]

P.T.O.