St. Vincent Pallotti
College of Engineering & Technology
(An Autonomous Institute)

Academic Year: 2022-23

CAT II - B.Tech

Course: -	**Object-Oriented Programming**	**Course Code: -**	DS402T
		Max. Marks: -	35
Semester: -	**IV**	**Duration: -**	1 hour 30 mins
Program: -	**Computer Science & Engg. (Data Science)**	**Date of Paper**	13/07/2023

Instructions to Candidate –
1) Question No. 1 is compulsory.
2) Solve Que. No. 02 OR Que. No. 03
3) Solve Que. No. 04 OR Que. No. 05
4) Solve Que. No. 06 OR Que. No. 07
5) All Questions carry marks as indicated
6) Programs can be written using any OOP language (C++, Java, etc.)
7) Credit reserved for neat indented programs and to-the-point answers.

Q.1 d)
- Each try block must have its own catch block to handle exceptions specifically for that try block code.
- This ensures exceptions are handled appropriately.
- If we combine multiple ... block it may lead to improper handling of exceptions

Que. No.	Description of Question	Marks	[CO]	[BTL]
Que.1(a)	What is the significance of the keyword "virtual" in OOP?	02	3	1
Que.1(b)	Define Pure Virtual Function.	02	4	1
Que.1(c)	Give an example of Static Binding in OOP.	02	4	1
Que.1(d)	Is it possible to handle multiple `try` blocks with a single `catch` block? Justify your answer. No it is not possible.	02	5	1
Que.1(e)	What is the difference between `throw` and `throws` in Java exception handling?	02	5	1
Que.2	Explain the Virtual Base Class with a suitable example.	05	3	2
	OR			
Que.3	Explain the Abstract class with a suitable example.	05	3	2
Que.4(a)	Illustrate the concept of Constructor Overloading with a programming example.	05	4	2
Que.4(b)	Justify the need for Virtual Destructor with a suitable programming example.	05	4	5
	OR			
Que.5	Design and implement a *class B* with three overloaded virtual functions. Derive a new *class D* from *B* and override one of the member functions. Also, write appropriate constructors for each class in the hierarchy. Make sure all Member Functions are	10	4	6

Q.1 e) Throw keyword is used to explicitly throw an exception.
It allows to create and throw an exception object to indicate error has occured.

Throws keyword is used to indicate the method might throw one or more exceptions.
It expects the caller to handle exceptions. caller must use try-catch block to handle exceptions.

	announcing themselves with appropriate messages. In *main()*, create an object of the *D* class and invoke all Member Functions of the base class through the derived class object. Analyze the behavior of the complete code and explain in short.			
Que.6(a)	Summarize the concept of try, catch, and throwing an exception.	05	5	2
Que.6(b)	Outline any three benefits of Exception Handling.	05	5	2
	OR			
Que.7	Design & implement a Stack data structure with exception handling to handle scenarios such as stack overflow and stack underflow using the following guidelines: 1. *Push* Operation: • When pushing an element onto the stack, you need to handle the possibility of a stack overflow. • To handle this, you can throw a custom exception, such as a `StackOverflowException`, when attempting to push an element onto a full stack. 2. *Pop* Operation: • When popping an element from the stack, you need to handle the scenario where the stack is empty, resulting in a stack underflow. • To handle this, you can throw a custom exception, such as a `StackUnderflowException`, when attempting to pop an element from an empty stack. 3. *Peek* Operation: • When peeking at the top element of the stack without removing it, you also need to handle the case where the stack is empty. • Like the pop operation, you can throw a `StackUnderflowException` when attempting to peek at an empty stack.	10	5	6

Course Outcomes - After Completion of this course student will -

CO1	Analyze and think in terms of object oriented paradigm during development of application
CO2	Apply the concept object initialization and destroy using constructors and destructors.
CO3	Develop application using the concept of inheritance and evaluate the usefulness.
CO4	Apply the concept of abstract class, polymorphism to implement compile time and runtime polymorphism.
CO5	Apply the concept of abstract class, polymorphism to implement compile time and runtime polymorphism.

Cat 1
Q.1 4) It identifies the overloaded functions of same name, different parameters.
It differentiates on basis of their method signature, which include function name and type or no of parameters

S/23/R

St. Vincent Pallotti College of Engineering & Technology
(An Autonomous Institute)
B. Tech Fourth Sem Computer Engineering / Computer Science and Engineering (Data Science)
End Semester Examination

Course: - Object oriented Programming **Course Code: -** CE404T/DS402T

Duration: - 3 hours **Max. Marks: -** 70

Instructions to Candidate – 1) Question No. 1 is compulsory.
2) Solve Q. No. 02 OR Q. No. 03
3) Solve Q. No. 04 OR Q. No. 05
4) Solve Q. No. 06 OR Q. No. 07
5) Solve Q. No. 08 OR Q. No. 09
6) Solve Q. No. 10 OR Q. No. 11
7) All Questions carry marks as indicated
8) Use of Non Programmable calculator is allowed.

Q. No.	Description of Question	Marks	[CO]	[BTL]
Q.1(a)	Differentiate between procedural programming language and object oriented language.	02	1	1
Q.1(b)	Explain how to create objects of a class with example.	02	1	1
Q.1(c)	Write about friends to a class.	02	2	2
Q.1(d)	Define constructor. List and explain the different types of constructors in C++.	02	2	1
Q.1(e)	Define the following terms, (i) Base Class(ii) Derived Class	02	3	2
Q.1(f)	Enlist the types of inheritances.	02	3	2
Q.1(g)	What is polymorphism? Explain the types of polymorphism.	02	4	1
Q.1(h)	Write a note on virtual destructors.	02	4	2
Q.1(i)	Define exception and explain in short about exception handling.	02	5	1
Q.1(j)	Discuss about exception objects.	02	5	2
Q.2(a)	Explain with example about UML class diagram.	05	1	2
Q.2(b)	Discuss access control of members of class.	05	1	2
Q.3(a)	Discuss the features of OOP language.	05	1	3
Q.3(b)	Write benefits of OOPS and also explain the applications of OOPS	05	1	3
Q.4(a)	Explain in details about the operator overloading with programming example.	05	2	2
Q.4(b)	Explain with example types of access specifiers.	05	3	1
Q.5(a)	Write about constructor, its purpose and advantages.	05	2	2
Q.5(b)	Explain about copy constructors.	05	3	3

Q.6(a)	Write a program to implement inheritance for the following statement: Accept and display data of one teacher and one student using object of class teacher & student derived from the base class Info.	05	3	2
Q.6(b)	Write a program to declare a class COLLEGE with members as college code. Derive a new class as STUDENT with members as studid. Accept and display details of student along with college for one object of student.	05	3	2
Q.7(a)	Explain the behavior of constructor calling in three different classes with multiple inheritance..	05	3	1
Q.7(b)	What is hybrid inheritance? Give one example and explain it.	05	3	1
Q.8(a)	"Polymorphism is implemented using function overloading." Justify the statement.	05	4	2
Q.8(b)	Write a program to declare a class Adder with data members as x and y. Initialize value of x and y with constructor. Calculate addition and display it using function "display".	05	4	4
Q.9(a)	What is destructor? Explain destructor with example. How many destructors can be defined in a single class?	05	4	3
Q.9(b)	Write a program to calculate area of circle and rectangle using the concept of function overloading.	05	4	3
Q.10(a)	Explain how to throw an exception in detail.	05	5	4
Q.10(b)	Explain the benefits of exception handling	05	5	2
Q.11(a)	Write a program to implement stack data structure. Use exception handling to check overflow and underflow conditions.	05	5	4
Q.11(b)	Explain the process of catching all exceptions.	05	5	4