COURSE TITLE	**DIGITAL TRANSFORMATION-I**
COURSE CODE	**05MF0203**
COURSE CREDITS	**4**

Objective:

1 To understand about the concept of OOPs.

2 To impart a thorough understanding about Python Constructor and creating the constructor in Python.

3 To understand about GUI using Tkinter.

4 To learn about Odoo and guided project.

5 To learn about process of adding products to the shopping cart in an Odoo Web-based shopping website.

Course Outcomes: After completion of this course, student will be able to:

1 Students can develop a solid understanding of Object-Oriented Programming (OOP) fundamentals in Python, including class creation, object instantiation, and deletion, distinguishing Object-oriented from Procedure-oriented programming languages, thereby enabling them to design and implement efficient, modular code structures for robust and scalable applications.

2 Students can deepen their Python expertise by mastering various topics such as constructors, both non-parameterized and parameterized, single and multi-level inheritance, multiple inheritance, method overriding, and overloading, thereby enhancing code organization and promoting reusability in their projects.

3 Students will explore the Tkinter GUI library in Python, covering components like Button, Frame, Label, Listbox, Menu, Menu Button, Message, Radiobutton, Message Box, Text, and Scrollbar, to develop interactive user interfaces, alongside learning MySQL database management and establishing Python-MySQL connectivity, culminating in the integration of Tkinter for frontend and MySQL for backend data storage and retrieval in graphical user interface application design.

4 Students will go through the process of signing up for Odoo Online, selecting the eCommerce App, configuring their website, adding products, setting up payment methods, configuring shipping methods, setting up taxes, customizing product pages, launching their website, and finally, managing their online store project.

5 Students with fundamental Object-Oriented Programming principles using Python, particularly focusing on encapsulation, inheritance, and polymorphism. Additionally, we will explore GUI development using Tkinter to create interactive applications. Furthermore, we intend to introduce the basics of the Odoo framework, providing an overview of its modular structure, development environment, enabling students to grasp the essentials of building business applications with Odoo.

Pre-requisite of course: Fundamental of Python Programming

DR. SUNIL LALCHAND BAJEJA
Digitally signed by (Name of HOD)

DR. R SRIDARAN
Digitally signed by (Name of Dean/ Principal)

Teaching and Examination Scheme

Theory Hours	Tutorial Hours	Practical Hours	ESE	IA	CSE	Viva	Term Work
3	0	2	50	30	20	25	25

Contents : Unit	Topics	Contact Hours
1	**Basics of OOPs and Classes** Python OOPs Concepts: Object-oriented vs. Procedure-oriented Programming languages., Python Class and Objects: Creating classes in Python.Creating an instance of the class. Deleting the object., Python Constructor: o Creating the constructor in Python. o Non-Parameterized Constructor., Python Parameterized Constructor.	11
2	**Advanced OOPs Concepts** Python Inheritance: o Single Inheritance. o Multi-Level Inheritance. o Multiple Inheritance. , Method Overriding. , Method Overloading.	11
3	**Introduction to GUI with Tkinter** Introduction of GUI Library Tkinter. , Uses of components: o Button, Frame. o Label, List Box, Menu. o Menu Button, Message, Radiobutton. o Message Box, Text, Scrollbar. MySQL Introduction and Python MySQL Connectivity.	11
4	**Guided Project with Tkinter and Odoo** Design a GUI application using Tkinter and MySQL. , Introduction to Odoo: o Sign Up for Odoo Online. o Select the eCommerce App. , Configure Your Website: o Add Products. o Configure Shipping Methods. o Set Up Other Configurations. o Customize Product Pages. , Launch Your Website and Manage Your Online Store.	12
Total Hours		**45**

Suggested List of Experiments:

DR. SUNIL LALCHAND BAJEJA
Digitally signed by (Name of HOD)

DR. R SRIDARAN
Digitally signed by (Name of Dean/ Principal)

Contents : Unit	Topics	Contact Hours
1	**Unit 1** 1. Write a Python program that illustrates the difference between object- oriented and procedure-oriented programming approaches by solving a simple problem using both paradigms. 2. Create a class in Python representing a car. Instantiate multiple car objects and perform operations like accelerating, braking, and changing gear on each object. 3. Write a Python program that defines a class representing a student. Include attributes such as name, age, and grade. Instantiate student objects and display their information. 4. Develop a Python program that defines a class representing a book. Create an instance of the book class and initialize its attributes such as title, author, and publication year. 5. Write a Python program that defines a class representing a file. Instantiate multiple file objects and delete each object one by one. 6. Create a Python class representing a bank account. Implement methods for depositing, withdrawing, and checking balance while encapsulating account details. 7. Develop a Python program that models a vehicle hierarchy. Implement classes for vehicles such as car, bike, and truck, inheriting common features from a parent class. 8. Write a Python program that demonstrates the difference between class variables and instance variables by defining a class representing a company and its employees. 9. Create a Python program that defines a class representing a product. Instantiate product objects and demonstrate accessing their attributes and methods. 10. Develop a Python program that showcases method overriding. 11. Write a Python program that demonstrates method overloading by defining a class with multiple methods having the same name but different parameters, and then invoking each method with different arguments. 12. Develop a Python program where you define a class representing a utility tool. Implement a static method within the class to perform a common calculation or operation. 13. Develop a Python program where you define a class representing a utility tool. Implement a static method within the class to perform a common calculation or operation. 14. Create a Python program that represents a shape hierarchy, with a parent class representing shapes and child classes representing specific shapes like circles and rectangles. Implement abstraction by defining abstract methods for calculating area and perimeter in the parent class. 15. Write a Python program that accesses class attributes and methods without creating an object by directly calling the class name followed by the method or attribute. 16. Implement a Python program that simulates an employee management system. Define a parent class for employees and child classes for different types of employees like full-time, part-time, and contract-based employees. 17. Develop a Python program that represents a computer system. Define classes for components like CPU, memory, and storage, and compose them together in a Computer class. 18. Create a Python project such as a simple game or a library management system, and demonstrate encapsulation by encapsulating sensitive data and exposing only necessary methods.	15

DR. SUNIL LALCHAND BAJEJA
Digitally signed by (Name of HOD)

DR. R SRIDARAN
Digitally signed by (Name of Dean/ Principal)

Suggested List of Experiments:

Contents : Unit	Topics	Contact Hours
2	**Unit 2** 1. Implement a Python program representing a shape hierarchy, where the parent class represents a shape and child classes represent specific shapes like circles and rectangles. Override the parent class's method to calculate the area for each specific shape. 2. Write a Python program that defines a class representing a vehicle. Implement a child class for each type of vehicle, such as car, bike, and truck, and add specific functionality to each child class. 3. Design a Python GUI using the Tkinter framework to create a form with three labels: "Printer ID", "Printer Model", and "Company Name". Add three text entry boxes (Entry widgets) corresponding to these labels and one button with the text "Submit". Upon clicking the "Submit" button, a message box should appear displaying the message "Your data is submitted". 4. Design a Python GUI using the Tkinter framework to create a form with three labels: "Employee ID", "Employee Name", and "Department". Include three text entry boxes (Entry widgets) for user input and one button with the text "Submit". When the "Submit" button is clicked, a message box should pop up showing the message "Your data is submitted" 5. Describe the main features and functionalities of Odoo Web for building a shopping website.	15
Total Hours		**30**

Textbook :

1 Programming Python: Powerful Object-Oriented Programming, Mark Lutz, O'Reilly Media, 2010

References:

1 Python, Python, Martin C. Brown, McGraw-Hill Education, 2001

2 Python Programming, Python Programming, Pearson, Pearson, 2016

3 Working with Odoo, Working with Odoo, Greg Moss, Packt Publishing, 2015

4 Core Python Programming, Core Python Programming, 3ed by R. Nageswara Rao, Dreamtech Press, 2017

Suggested Theory Distribution:

The suggested theory distribution as per Bloom’s taxonomy is as follows. This distribution serves as guidelines for teachers and students to achieve effective teaching-learning process

Distribution of Theory for course delivery					
Remember / Knowledge	**Understand**	**Apply**	**Analyze**	**Evaluate**	**Higher order Thinking / Creative**
10.00	20.00	25.00	25.00	10.00	10.00

DR. SUNIL LALCHAND BAJEJA

Digitally signed by (Name of HOD)

DR. R SRIDARAN

Digitally signed by (Name of Dean/ Principal)

Instructional Method:

1 Board Work
2 PPT
3 Demo

Supplementary Resources:

1 https://www.lio.io/digital-transformation
2 https://www.techrepublic.com/

DR. SUNIL LALCHAND BAJEJA
Digitally signed by (Name of HOD)

DR. R SRIDARAN
Digitally signed by (Name of Dean/ Principal)