

CODE	COURSE NAME	CATEGORY	L	T	P	CREDIT
ITT202	PRINCIPLES OF OBJECT ORIENTED TECHNIQUES	PCC	3	1	0	4

Preamble:

This course is intended to make the students capable of

1. Compare the capabilities of Object oriented and Procedure oriented programming languages.
2. Model the problem scenarios using object oriented concepts and UML.
3. Develop robust programs by optimally utilising the capabilities JAVA programming language.

Prerequisite: Programming Concepts

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

CO No	Course Outcomes	Bloom's Category
CO 1	Describe the specific capabilities of Object-Oriented paradigm compared to procedure oriented paradigm	Understand
CO 2	Explain the use of object oriented concepts to analyse the given problem.	Understand
CO 3	Describe the different inheritance features available in Java	Understand
CO 4	Construct robust programs using Exception Handling	Apply
CO 5	Construct applets utilising multithreading, event handling and Graphical User Interface, also model the problem scenarios using UML diagrams.	Apply

Mapping of course outcomes with program outcomes

COs	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12
CO 1	2	3	3	-	2	-	-	-	-	-	-	1
CO 2	2	3	3	-	2	-	-	-	-	-	-	-
CO 3	2	3	3	-	2	-	-	-	-	-	-	-
CO 4	2	3	3	-	2	-	-	-	-	-	-	-
CO 5	2	3	3	-	2	-	-	-	-	-	-	-

3/2/1: high/medium/low

Assessment Pattern

Bloom's Category	Continuous Assessment Tests		End Semester Examination
	1	2	
Remember			
Understand	25	25	50
Apply	25	25	50
Analyse			
Evaluate			
Create			

Mark distribution

Total Marks	CIE	ESE	ESE Duration
150	50	100	3 hours

Continuous Internal Evaluation Pattern:

Attendance	: 10 marks
Continuous Assessment Test (2 numbers)	: 25 marks
Assignment/Quiz/Course project	: 15 marks

End Semester Examination Pattern: There will be two parts; Part A and Part B. Part A contain 10 questions with 2 questions from each module, having 3 marks for each question. Students should answer all questions. Part B contains 2 questions from each module of which student should answer any one. Each question can have maximum 2 sub-divisions and carry 14 marks.

Course Level Assessment Questions

Course Outcome 1 (CO1):

1. Compare and contrast the implementation of data abstraction in procedure oriented and object oriented language.
2. Explain how encapsulation helps in data security. Justify your answer by comparing the scenario in procedure oriented languages.
3. Describe TWO features of object oriented programming languages that promote code reuse.

Course Outcome 2 (CO2):

1. Describe about the statement “String is a primitive data type or not in C++.
2. Describe the use of ‘static’ functions in C and Java.

Course Outcome 3(CO3):

1. Explain the difference between the object oriented design concepts of generalisation and specialisation, and describe how these relate to the inheritance feature in object oriented programming languages.
2. Describe how is-a and has-a inter-class relationships may be implemented in object oriented programming, giving code examples to support your answer.
3. What is multiple inheritance? Discuss how multiple inheritance is implemented in Java.

Course Outcome 4(CO4):

1. Experiment with runtime and compile time errors. Would you rather have an error discovered at run time or compile time?
2. Experiment with out of bound exception with example code.
3. Build java programs using following constructs.
 - a) try { }
 - b) catch { }
 - c) throw()

Course Outcome 5(CO5):

1. Experiment with multithreaded applets.
2. Experiment with inter-process communication mechanism in java.
3. Design a class diagram of the following System: Vending Machine. A vending machine sells small, packaged, ready to eat items (chocolate bars, cookies, candies, etc.). Each item has a price and a name. A customer can buy an item, using a smart card (issued by the vending machine company) to pay for it. No other payment forms (i.e. cash, credit card) are allowed. The smart card records on it the amount of money available. The functions supported by the system are: Sell an item(choose from a list of items, pay item, distribute item)Recharge the machine Set up the machine (define items sold and price of items)Monitor the machine (number of items sold, number of items sold per type, total revenue)The

system can be used by a customer, a maintenance employee (who recharges items in the machines), an administrator (who sets up the machine).

Model Question paper

Course Code: ITT202

Course Name: PRINCIPLES OF OBJECT ORIENTED TECHNIQUES

Max.Marks:100

Duration: 3 Hours

PART A

Answer all Questions. Each question carries 3 Marks

1. Explain the use of 'static' functions in C and Java.
2. Comment on the statement "String is a primitive data type or not in C++.
3. Explain the use of 'this' keyword in Java.
4. Write a short note on implementation of "write once, run anywhere" concept in Java
5. Explain encapsulation using class in java.
6. Write short note on inner calss in Java.
7. Explain the use of dynamic method dispatch.
8. Differentiate between 'throw', 'throws', and 'Throwable'.
9. Explain event handling model with diagram.
10. Write short note on sand box security model for applets.

Part B

Answer any one Question from each module. Each question carries 14 Marks

- 11.a) Compare and contrast the implementation of data abstraction in procedure oriented and object oriented language. 7Marks
- b) Discuss how encapsulation helps in data security. Justify your answer by comparing the scenario in procedure oriented languages. 7 Marks

OR

12. a) Write a short note on garbage collection in java. 6 Marks
- b) Describe with example, TWO features of object oriented programming languages that promote code reuse. 8 Marks

13. a) Compare and contrast overriding and overloading with examples.

6 Marks

b) With an example, discuss the use of passing objects as parameters.

8 Marks

OR

14.a) Discuss the uses of 'static' keyword with example scenarios. 7 marks

b) Examine the use of 'final' keyword in the context of access control.

7Marks

15.a) Explain the difference between the object oriented design concepts of generalisation and specialisation, and describe how these relate to the inheritance feature in object oriented programming languages.

6 marks

b) Describe how is-a and has-a inter-class relationships may be implemented in object oriented programming, giving code examples to support your answer.

8 Marks

OR

16.a) What is multiple inheritance? Discuss how multiple inheritance is implemented in Java.

7 marks

b) Compare and contrast the usage of abstract class and interface in Java. Give examples of each.

7 Marks

17.a) Discuss the difference between runtime and compile time errors. Would you rather have an error discovered at run time or compile time?

8 Marks

b) Explain out of bound exception with example code.

6 marks

OR

18. a) Explain the use of following constructs in Java with example.

i)try { }

ii)catch { }

iii)throw()

6 Marks

b) Elaborate on the interprocess communication mechanism in java.

8 Marks

19. a) Discuss how to implement a multithreaded applet with an example.

6 Marks

b) Explain with an example the event model in Java.

8 Marks

OR

20. a) Differentiate between static and dynamic models in UML. 4 Marks

b) Draw a class diagram of the following System: Vending Machine. A vending machine sells small, packaged, ready to eat items (chocolate bars, cookies, candies, etc.). Each item has a price and a name. A customer can buy an item, using a smart card (issued by the vending machine company) to pay for it. No other payment forms (i.e. cash, credit card) are allowed. The smart card records on it the amount of money available. The functions supported by the system are: Sell an item (choose from a list of items, pay item, distribute item) Recharge the machine Set up the machine (define items sold and price of items) Monitor the machine (number of items sold, number of items sold per type, total revenue) The system can be used by a customer, a maintenance employee (who recharges items in the machines), an administrator (who sets up the machine).

10 Marks

Syllabus

Module 1	No. of Lectures
Object-Oriented Programming vs Procedure-Oriented Programming, Procedural Languages - The Object-Oriented Approach - Characteristics of Object-Oriented Languages Objects –Classes – Inheritance – Reusability - Creating New Data Types - Polymorphism and Overloading ,Object oriented concepts in Java -Java Overview: Java virtual machine, data types, operators, control statements, Classes fundamentals, objects, methods, constructors, this keyword, Garbage collection	10 hours
Module 2	
Overloading Methods, Overloading Constructors, Using Objects as Parameters, Call by value and Call by reference, Access control, use of static and final keywords, Nested and Inner classes	8 hours
Module 3	
Derived Class and Base Class, Usage of super keyword, Creating a Multilevel Hierarchy, Method Overriding, Dynamic Method Dispatch, Using Abstract Classes, Using final with Inheritance, Definition and application of Packages and Interfaces	8 hours
Module 4	
Fundamentals of exception handling, Exception Types, Using try and catch, throw, throws, finally, Java's Built-in Exceptions, Creating Exception subclasses, the Java Thread Model, Creating a Thread, Creating Multiple Threads, Thread Priorities, Synchronization, Interthread	9 hours

Communication	
Module 5	
Event Handling-delegation event model, event classes, sources, listeners. String class - basics. Applet basics and methods, AWT- working with frames, graphics, color, font. AWT Control fundamentals. Swing overview, Introduction to Object Oriented Modelling ,Unified Modeling Language, UML class diagram, Use-case diagram, Familiarisation of UML tools, Case study	10 hours

Text Books

1. Herbert Schildt, Java: The Complete Reference, 8/e, Tata McGraw Hill, 2011.
2. Bahrami A., Object Oriented Systems Development using the Unified Modeling Language, McGraw Hill, 1999.

Reference Books

1. Flanagan D., Java in A Nutshell, 5/e, O'Reilly, 2005.
2. Sierra K., Head First Java, 2/e, O'Reilly, 2005.
3. Balagurusamy E., Programming JAVA a Primer, 5/e, McGraw Hill, 2014.
4. Barclay K., J. Savage, Object Oriented Design with UML and Java, Elsevier, 2004.
5. James Rumbaugh., Unified Modeling Language Reference Manual, Addison-Wesley Professional, 2005

Course Contents and Lecture Schedule

No	Topic	No. of Lectures
1	Object Oriented concepts	10 Hours
1.1	Object-Oriented Programming vs Procedure-orientated Programming, Procedural Languages - The Object-Oriented Approach - Characteristics of Object-Oriented Languages	3 hours
1.2	Objects –Classes – Inheritance – Reusability - Creating New Data Types - Polymorphism and Overloading	3 hours
1.3	Object oriented concepts in Java -Java Overview: Java virtual machine, data types, operators, control statements, Classes fundamentals, objects, methods, constructors, this keyword, Garbage collection	4 hours
2	Method overloading	8 Hours

2.1	Overloading Methods, Overloading Constructors, Using Objects as Parameters	4 hours
2.2	Access control, use of static and final keywords, Nested and Inner classes	4 hours
3	Inheritance	8 Hours
3.1	Derived Class and Base Class, Usage of super keyword, Creating a Multilevel Hierarchy, Method Overriding	4 hours
3.2	Dynamic Method Dispatch, Using Abstract Classes, Using final with Inheritance, Definition and application of Packages and Interfaces	4 hours
4	Exception handling and Multithreaded Programming	9 Hours
4.1	Fundamentals of exception handling, Exception Types, Using try and catch, throw, throws, finally, Java's Built-in Exceptions, Creating Exception subclasses.	4 hours
4.2	The Java Thread Model, Creating a Thread, Creating Multiple Threads, Thread Priorities, Synchronization, Interthread Communication	5 hours
5	Event Handling, AWT and UML	10 Hours
5.1	Event Handling-delegation event model, event classes, sources, listeners. String class – basics. Applet basics and methods	3 hours
5.2	AWT- working with frames, graphics, color, font. AWT Control fundamentals. Swing overview	3 hours
5.3	Introduction to Object Oriented Modelling-Unified Modelling Language, UML class diagram, Use-case diagram, Familiarisation of UML tools, Case study	4 hours