

Reg. No. :

--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

Question Paper Code : 21388

B.E./B.Tech. DEGREE EXAMINATION, NOVEMBER/DECEMBER 2015

Fifth Semester

Computer Science And Engineering

CS 2305/CS 55/10144 CS 506 – PROGRAMMING PARADIGMS/PROGRAMMING
PARADIGMS WITH JAVA

(Regulations 2008/2010)

Time : Three hours

Maximum : 100 marks

Answer ALL questions.

PART A — (10 × 2 = 20 marks)

1. Mention the various access specifiers supported by object oriented programming.
2. Write a simple JAVA program to find the given number is prime or not
3. What is meant by object cloning?
4. Define dynamic binding
5. List the various mouse events supported by JAVA.
6. State the role of layout manager. Which layout is default in JAVA?
7. What is meant by generic programming?
8. Mention the elements of stack trace in generic programming.
9. State the properties of thread.
10. Why is synchronization required in thread?

PART B — (5 × 16 = 80 marks)

11. (a) Explain the fundamentals of OOPs. Discuss the purpose of finalize method with example.

Or

- (b) What is meant by constructor? Discuss the types of constructors with example.
12. (a) What is abstract class? State the purpose of it. Illustrate the concept of inner class with example.

Or

- (b) Describe the concept of reflection and interfaces with example.
13. (a) What is meant by event handling? Develop a simple calculator using mouse events that restrict only addition, subtraction, multiplication and division.

Or

- (b) Explain the controller design pattern and components of swing briefly.
14. (a) State the motivations of generic programming. Explain the generic class and methods with example.

Or

- (b) What do you mean by error handling? Describe the throwing and catching exceptions supported in generic programming. Give example.
15. (a) What is thread? Discuss the types of threads. Write a program for reader and writer problem using thread in JAVA.

Or

- (b) Explain the two types of thread implementation supported by JAVA. Give examples. Write note on interrupting threads.