

**2018**

**M.Sc.**

**2nd Semester Examination**

**COMPUTER SCIENCE**

**PAPER—COS-206**

**Subject Code—26**

**(PRACTICAL)**

**Full Marks : 50**

**Time : 3 Hours**

*the figures in the right-hand margin indicate full marks.*

*Candidates are required to give their answers in their own words as far as practicable.*

*Illustrate the answers wherever necessary.*

**Answer any two questions : 2×15**

- 1. Write a Java program to display the  $n$ -th Fibonacci number. Input the  $n$  through Command line.**
- 2. Write a Java Program to find the roots of a quadratic equation. Assign the values of  $a$ ,  $b$  and  $c$  of the equation in the Program itself.**

*(Turn Over)*

3. Write a Java program to create a Package called "Arithmetic" that contains Method to deal all arithmetic operation.
4. Write a program for conductor overloading.
5. Write a Java Program for an interface that has at least two methods and two data members.
6. Write a Java program to find the sum of following series :  
 $1+2+4+7+11+16+\dots$  (Apply Keyboard connection).
7. Write a Java program for Multi level inheritance using Supper().
8. Write a Java program for checking of String Palindrome.
9. Write a Java program to implement the Method Overriding.
10. Write a Java program for Handling Excertion.
11. Write a Java program to calculate factorial of any number.

12. Write a Java program to implement applet.
13. Write a Java program to implement Abstract class.
14. Write a Java program to find GCD and LCM of any two numbers.
15. Create a class Prime. Add two functions: checkPrime and printPrimeSeries. The checkPrime function checks whether a given number is a prime number or not. The printPrimeSeries function prints all prime numbers within a given range. When a number is passed as command line argument, the program checks whether the given number is prime. Add other necessary variables and functions in the class.
16. Create a class GCDLCM. Add two functions getGCD and getLCM in the class. The function getGCD returns the GCD of numbers given as arguments. Similarly, the getLCM function returns the LCM of numbers given as arguments. When the numbers are passed as command line arguments, the program prints both GCD and LCM of the given numbers. Add other necessary variables and functions in the class.

17. Create a class `Fibonacci`. Add two functions: `getNthFibonacci`, `printFibonacciSeries` in the class. The `getNthFibonacci` function returns `nth` Fibonacci number. The function `printFibonacciSeries` prints Fibonacci series within a range. Also print only the prime Fibonacci numbers or the nonprime Fibonacci numbers using the `printFibonacciSeries` function. Try to reuse any function whenever possible. Add other necessary variables and functions in the class.
18. Create a class `Factorial`. Add two functions: `getFactorial`, `printFactorialSeries`. The `getFactorial` function returns factorial of a given number. The `printFactorialSeries` function prints all factorials of numbers within a range. When a number is passed as command line argument, the program prints the factorial of the given number. Add other necessary variables and functions in the class.
19. Create a class `Armstrong`. Add two functions: `checkArmstrong`, `printArmstrongNumbers`. The `checkArmstrong` function checks whether a given number is an Armstrong number or not. The `printArmstrongNumbers` function prints all Armstrong numbers within a range. When a number is passed as

command line argument, the program checks whether the given number is Armstrong or not. Add other necessary variables and functions in the class.

20. Write a multithreaded program to show thread synchronization in Java. Create at least two threads in your application. Create necessary classes and add variables and member functions in the classes.
21. Write a program to implement an interface and extend an abstract class. The interface and the abstract class should have at least one function. Add other necessary variables and functions in the class to show the usability of an interface and an abstract class.
22. Create a class Shape. Create three subclasses Triangle, Rectangle and Circle. Add at least two functions and necessary variables in these classes. Implement method overloading and method overriding using these classes.
23. Write an Applet program in Java. Implement all necessary functions of the applet to show their functionality. Draw any two figures and some text in the applet using different colors.

24. Write a program to show how Java handles exceptions in an application. Use try, catch and finally block in the program.

**[Practical Note Book : 5 Marks**

**Viva Voce : 15 Marks]**

---