Reg. No. \_\_\_\_\_\_\_\_\_\_\_\_



**End Semester Examination – Nov / Dec – 2019**

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
| **Code :** | **17CS2017** | **Duration :** | **3hrs** |
| **Sub. Name :** | **PROGRAMMING IN JAVA** | **Max. Marks :** | **100** |

**ANSWER ALL QUESTIONS (5 x 20 = 100 Marks)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Q. No.** | **Sub Div.** | **Questions** | **Course**  **Outcome** | **Marks** |
| 1. | a. | Elaborate any five features of Java. | CO2 | 10 |
| b. | Write a program to reverse the digits of a given number. | CO2 | 4 |
| c. | Describe any three relational operators with suitable examples. | CO2 | 6 |
| OR | | | | |
| 2. | a. | Explain the various selection and iteration control statements with sample code. | CO2 | 12 |
| b. | Write a program to store the given N numbers in an array and perform the following operations.  (i) Find and display the average  (ii) Search the given number in the array and display.  “Available” if it exists and “Not Available” if it does not exist. | CO2 | 8 |
|  |  |  |  |  |
| 3. | a. | Develop a java program to design a class for “Smartphone” with model\_name, year, cost as data members. Include constructors and setter/getter methods. Add another class “Main” which creates two objects, initialize the data members and display the details through getter methods. | CO4 | 7 |
|  | b. | List and elaborate on four types of inheritance with sample programs. | CO1 | 8 |
|  | c. | Compare and contrast the method overloading and overriding with example programs. | CO1 | 5 |
| OR | | | | |
| 4. | a. | Discuss the usage of packages and illustrate the various access specifiers in packages with examples. | CO1 | 8 |
|  | b. | Create an interface called “Engine” with set of methods changeGear(), speedUp(), applyBrakes(). Create two classes “Car” and “Bike” with properties gear, speed which implement the interface functionalities. Include a “Main” class to demonstrate the accessing of those interface methods. | CO4 | 12 |
|  |  |  |  |  |
| 5. | a. | Examine the two ways of creating multiple threads with illustrations of sleep() and join() methods. | CO5 | 10 |
|  | b. | Create a program to demonstrate an enumeration with data, methods and constructor. | CO2 | 10 |
| OR | | | | |
| 6. | a. | Create a user defined exception to handle “insufficient balance” and “deposit beyond limit” situations. Use the in Account class and demonstrate same. | CO1 | 10 |
|  | b. | List the various types of wrapper classes and mention the significance of auto boxing/unboxing with sample code. | CO2 | 5 |
|  | c. | Design a lambda expression and functional interface to check the given ‘name’ of a person is starting with ‘S’ or not. | CO2 | 5 |
| 7. | a. | Describe the following methods of String Buffer with example code snippets and output justifications.  i) insert() ii) delete() iii) reverse() iv) append() v) delete CharAt() | CO2 | 10 |
|  | b. | Differentiate Byte Streams and Character Streams. | CO5 | 4 |
|  | c. | Create a program that store the names of 5 students in an Array List collection and display the same using Iterator. | CO5 | 6 |
| OR | | | | |
| 8. | a. | Summarize the various character extraction methods of String class with sample code. | CO2 | 5 |
| b. | Assume a file “xyz.txt” with textual contents. Write a program to read and display the entire contents of the file using File I/O. | CO5 | 5 |
| c. | Explain TCP Socket programming model with client and server implementations. | CO6 | 10 |
|  | |  |  |  |
|  | | **Compulsory:** |  |  |
| 9. | a. | Develop GUI program for Dollar to Rupee Convertor using Java Swing API as given below. | CO5 | 10 |
|  | b. | Discuss any four Layout Managers with neat sketch and suitable sample programs. | CO5 | 10 |

**Outcome:**

The student will be able to.

1. identify the necessary attributes and methods of an object, hierarchical classification of classes, and handle necessary exceptions.
2. infer object-oriented programming constructs, control statements and java library.
3. show proficiency in debugging.
4. model prototype for the real time application.
5. develop desktop application using multi-threading, collections, IO concepts, GUI to solve real-time problems.
6. design distributed applications using network concepts to solve distributed collaborative problems.