**Mobile phones are banned**

**M S Ramaiah Institute of Technology**

**Department of Information Science and Engineering**

|  |  |  |  |
| --- | --- | --- | --- |
| **Term:** | 28/07/2017 to 20/12/2017 | **Course Code:** | IS35B |
| **Course:** | Object Oriented Programming with Java | **Semester:** | 3 |
| **CIE:** | Test 2 | **Max Marks:** | 30 |
| **Date:** | **04/11/17** | **Time:** | **9 :30 –10:30 AM** |

**Portions for Test:** Lecture Nos. from 1 to 17 as per lesson plan

**Instructions to Candidates:** Answer any **Two** out of Three questions.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Sl No** | **Questions** | **Marks** | **Blooms Level** | **CO** |
| **1a.** | Using appropriate code-snippets explain what you understand by  ‘Dynamic Method Dispatch’? Why do we need the concept of ‘Static and Dynamic Method Overriding’? | **6** | **Un**  **L2** | **CO3** |
| **1b.** | There is a class called 'Tourist' which contains attributes such as ‘name’, 'gender' and 'age'. An unrelated class called 'MakeMyTrip' contains attributes called ‘dateOfJourney’, 'departurePoint' and ‘arrivalPoint’. It also has a member function called 'makeIternary', which displays the details of the tourist along with the departure, arrival destinations and date of journey. Write a program with main function to demonstrate this. | **9** | **An**  **L4** | **CO2** |
| **2a.** | A base class called 'Pizza' contains a string attribute 'item\_name’ and two number attributes 'price' and 'qty'. It also has a method called 'calculate' that returns the total price to be paid by the user. You must not allow the logic of 'calculate' to be changed by the sub-classes of 'Pizza'. Also you must ensure that objects of ‘Pizza’ are never created. How will you ensure these? Call 'calculate' through a sub-class of ‘Pizza’ called ‘CheesePizza’. Make assumptions for attributes of this sub-class. Write the complete implementation with main function. | **10** | **An**  **L4** | **CO3** |
| **2b.** | Using appropriate code-snippets explain the two uses of ‘super’ with inheritance. Why do you need these concepts? | **5** | **Un**  **L2** | **CO2** |
| **3a.** | There is a class called 'Game' which contains a method called 'gameDetails'. It has a sub-class called 'IndoorGame' containing the method 'gameDetails' which is overridden by it. Objects of 'Game' must be able to access the method 'gameDetails' of 'IndoorGame'. Which concept will you apply here? Explain briefly with code-snippets. | **5** | **Ap**  **L3** | **CO3** |
| **3b.** | With any code-example of your choice explain the **use** and **need** for ***any two*** of the following:  **(i)** Command-Line Arguments  **(ii)** Overloading and Ambiguities of Variable-Length Arguments.  **(iii)** Static and Non-Static Inner Class | **10**  **(i)5**  **(ii)5**  **(iii)5** | **Ap**  **L3** | **CO2** |

R- Remember, U-Understand , Ap- Apply and An-analyze.