Object Oriented Programming (PCC-CS593)
2025

| | |
|---|---|
| ROLL | |
| NAME | |

## ASSIGNMENT: III

| Sl. No | Program Listing | Date |
|---|---|---|
| 1 | Create a class named Rectangle that contains data fields for height, width and a method named getArea(). Create a child class named Cuboid. Cuboid contains an additional data field named depth, and a getArea() method that overrides the parent method.<br>Write an application that read the data from the input file named “input.txt” and instantiates Square object and Cuboid object and display the surface areas of the objects. Save files as Rectangle.java, Cuboid.java, and Demo.java | |
| 2 | Create an abstract class GeometricObject which will have two variables fillColor and borderColor. In the constructor setting the fillColor as “white” and the borderColor as “black” as default values. Write getter methods getFillColour() and getBorderColor() to return the values. The class should have two abstract methods findArea() and findCircumference(). Write a subclass of GeometricObject called Triangle which will be able to calculate area and circumference for a triangle. | |
| 3 | Consider Account class of Assignment II. Remove the account type from Account class. Create a child class SavingsAccount, contains an additional data field named interestRate and a computeInterest() method that compute the interest and returns. Overrides the deposit() and withdraw() methods, and toString method to return object as a string. Initial balance should be Rs 500. Imposed fine Rs.10 when balance goes under Rs 500.<br>Create another child class CurrentAccount, contains an additional data field named overDraftLimit and overrides the deposit() and withdraw() methods, and toString method to return object as a string. Initial balance should be Rs 5000. Imposed fine Rs.50 when balance goes under overdraft limit.<br>Create TestBankingSystem class to demonstrate the functioning of the above. Write the account details to the output file. | |
| 4 | Create a generic class named Stack using array, within the Stack class, write constructor, boolean pop() deletes the element on the top of the stack, boolean push(E e) pushes the element e onto the stack, T peek() returns the element on the top of the stack, boolean isEmpty(): check whether the stack is empty, boolean isFull() check whether the stack is full, void ensureCapacity() Increase the stack capacity by doubling the size. | |
| 5 | Create a LinkedList class, write a method insert that inserts integers at the end of the list. Write method that searches through the linked list and returns whether or not a specific number exists in the list, a method that remove a node from the linked list. Don’t allow duplicate elements. | |