Lab 2

1. WAP that reads inputs from user and writes it to the file test.txt until user enters bye. Then read and display the content of file.
2. Write a Python program to Merge two text files (file1.txt and file2.txt) into a single file named merged.txt.
3. Write a Python program to Read a log file named ‘server.log’ Extract all lines containing the word "ERROR" and write them to a new file named errors.log.
4. Write a Python program to Read a text file named input.txt Remove all blank lines from the file and convert text into uppercase and write the cleaned content to a new file named output.txt.
5. WAP that reads data from emp.csv file and displays content of column 0, 1, and 3.
6. WAP that reads columns 1,2, and 3 from emp.csv file and converts them into list of tuples where elements of the tuple are from column 1, 2, and 3 respectively.
7. WAP to handle divide by zero and invalid array index exceptions using try….except…..finally blocks.
8. WAP to create an abstract class named Person with data members Id, name, and age and abstract methods getData() and display(). Derive class Employee from person and add member variable salary. Again derive another class customer from person and add member variable credit rating. Finally create objects of Employee and Customer and read and display their details.
9. WAP to create class Customer with data members Cid, Cname, and Address. Define constructor to initialize objects of the class and display member function to display object data. Create another class Employee with data memebrs Eid, Ename, and Post. Again define constructor to initialize objects of the class and member function display to display object data. Derive a class named EmpCustomer from above two classes. Add member variable rating, constructor, and display member function in the class. Finally create object of EmpCustomer class and read and display object details.
10. WAP to create class Account with data members account number, account name, and balance. Add members function for reading and displaying data. Derive class saving account from account class. Add class variable minimum balance in the class. Define member functions deposit, withdraw, transfer, displayMinBalance (non-instance member function), and verify (static member function) in the class. Method verify should verify the amount at the time of withdraw and transfer operation. Finally, create an object of Saving account class and demonstrate all operations defined in the class.