Python Programming

Assignment – 2

1. Write a Python program to create a class named **Student** with attributes *name*, *age* and *grade*. Implement a method to display the student's information in a formatted string.
2. Write a Python program to create a class named **Animal** with an abstract method named *sound*. Implement subclasses for different animals, such as Dog, Cat, and Cow, and override the sound method for each subclass.
3. Write a Python program to create a class named **Rectangle** with attributes *length* and *width*. Implement methods to calculate the area and perimeter of the rectangle. Also, implement a method to compare two rectangles based on their area.
4. Write a Python program to create a class named **BankAccount** with attributes *owner* and *balance*. Implement methods to deposit, withdraw, and check the balance of the account. Also, implement a method to transfer money from one account to another.
5. Write a Python program to create a class named **Employee** with attributes *name*, *id*, and *salary*. Implement methods to get and set the salary of the employee. Also, implement a class method to calculate the average salary of all employees.
6. Write a Python program to create a class named **Car** with attributes *model*, *color* and *price*. Implement methods to start, stop, and accelerate the car. Also, implement a static method to count the number of cars created.
7. Write a Python program to create a class named **Book** with attributes *title*, *author* and *price*. Implement methods to display the book's information and apply a discount on the price. Also, implement a subclass named **EBook** with an additional attribute format. Override the display method for the **EBook** subclass.
8. Write a Python program to create a class named **Person** with attributes *name*, *age* and *gender*. Implement methods to greet and introduce the person. Also, implement multiple inheritance by creating two subclasses named **Student** and **Teacher** that inherit from **Person** and have additional attributes like *course* and *subject* respectively.
9. Write a Python program to create a class named **Calculator** with methods for basic arithmetic operations like add, subtract, multiply, and divide. Also, implement operator overloading by defining special methods like **add**, **sub**, **mul** and **truediv** for the **Calculator** class.
10. Write a Python program to create a class named **Stack** with methods for pushing and popping elements from the stack. Also, implement an iterator for the **Stack** class that returns the elements in LIFO (last in first out) order.