OOPJ Assignment

1. Create an abstract class “Publication” with data members ‘noOfPages’, ‘price’, ‘publisherName’ etc. with their accessor/modifier functions. Now create two sub-classes “Book” and “Journal”. Create a class Library that contains a list of Publications. Write a main() function and create three Books and two Journals, add them to library and print the details of all publications.

2. Write a class for “Account” containing data members ‘accountNumber’, ‘holderName’, ‘balance’ and add constructors and necessary accessor/modifier functions for these data members. Now create two class “SavingsAccount” and “CurrentAccount” extending from this class. SavingsAccount will have a member variable ‘interestRate’ and member function ‘calculateYearlyInterest’. Write another class “Manager” that contains a list Account. Also write a main() function to create an instance of Manager class. Add two SavingsAccount and three CurrentAccount to Manager. Calculate interest of each SavingsAccount. Print the details of all accounts.

3. Implement a class for a “Person”. Person has data members ‘age’, ’weight’, ‘height’, ‘dateOfBirth’, ‘address’ with proper reader/write methods etc. Now create two subclasses “Employee” and “Student”. Employee will have additional data member ‘salary’, ‘dateOfJoining’, ‘experience’ etc. Student has data members ‘roll’, ‘listOfSubjects’, their marks and methods ‘calculateGrade’. Again create two sub-classes “Technician” and “Professor” from Employee. Professor has data members ‘courses’, ‘listOfAdvisee’ and their add/remove methods. Write a main() function to demonstrate the creation of objects of different classes and their method calls.

4. Create a base class “Automobile”. An Automobile contains data members ‘make’, ‘type’, ‘maxSpeed’, ‘price’, ‘mileage’, ‘registrationNumber’ etc. with their reader/writer methods. Now create two sub-classes “Track” and “Car”. Track has data members ‘capacity’, ‘hoodType’, ‘noOfWheels’ etc. Car has data members ‘noOfDoors’, ‘seatingCapacity’ and their reader/writer methods. Create a main() function to demonstrate this.

5. Implement the classes for the following inheritance hierarchies. Create an interface “Shape” that contains methods ‘area’, ‘draw’, ‘rotate’, ‘move’ etc. Now create two classes “Circle” and “Rectangle” that implement this ‘Shape’ interface and implement the methods ‘area’, ‘move’, etc. Write a main() function to create two “Circle” and three “Rectangle”,then move them. Print the details of circles and rectangles before after moving them.

6. A bookshop maintains the inventory of books that are being sold at the shop. The list includes details such as author, title, publisher, cost and stock position. Whenever a customer wants a book, the sales person inputs the title and author and the system searches the list and displays whether it is available or not. If it is not, an appropriate message is displayed. If it is, then the system displays the book details and details and requests for the number of copies required. If the required copies are available, the total cost of the requested copies is displayed, otherwise the message “requires copies not in stock” is displayed. Design a system using a class called “Book” with suitable member methods and constructors.

7. Create a class called Author is designed as follows:

It contains: • three private instance variables: name (String), email (String), and gender (char of either ‘m’ or ‘f’). • One constructor to initialize the name, email and gender with the given values.

And, a class called Book is designed as follows: It contains: • Four private instance variables: name (String), author (of the class Author you have just created), price (double), and qtyInStock (int). Assuming that each book is written by one author. • One constructor which constructs an instance with the values given. • Getters and setters:

getName(), getAuthor(), getPrice(), setPrice(), getQtyInStock(), setQtyInStock(). Again there is no setter for name and author. Write the class Book (which uses the Author class written earlier). Try:

1.Printing the book name, price and qtyInStock from a Book instance. (Hint: aBook.getName())

2. After obtaining the “Author” object, print the Author (name, email & gender) of the book.