**BITS, Pilani - Hyderabad Campus**

**Object-Oriented Programming (CS F213)**

**Extra Practice Problems (20-09-2021 to 26-09-2021)**

**Topics: Inheritance and Polymorphism**

1. Create a class with a method that prints "This is parent class" and its subclass with another method that prints "This is child class". Now, create an object for each of the class and call  
   1 - method of parent class by object of parent class  
   2 - method of child class by object of child class  
   3 - method of parent class by object of child class

Now declare the method of the parent class as private and then repeat the first two operations (You will get error in the third).

1. Create a class named 'Shape' with a method to print "This is This is shape". Then create two other classes named 'Rectangle', 'Circle' inheriting the Shape class, both having a method to print "This is rectangular shape" and "This is circular shape" respectively. Create a subclass 'Square' of 'Rectangle' having a method to print "Square is a rectangle". Now call the method of 'Shape' and 'Rectangle' class by the object of 'Square' class.
2. A class called circle is designed with Two private instance variables: radius (of the type double) and color (of the type String), with default value of 1.0 and "red", respectively. Two overloaded constructors - a default constructor with no argument, and a constructor which takes a double argument for radius. Two public methods: getRadius() and getArea(), which return the radius and area of this instance, respectively.

Compile "Circle.java". Can you run the Circle class? Why?

Let us write a *test program* called TestCircle which uses the Circle class and display radius and area.

1. Create a class called Employee. This class will be a derived class of the class person. An employee record has an employee's name (inherited from the class Person), an annual salary represented as a single value of type double, a year the employee started work as a single value of type int.

IF employee joins 10 years before 2020 then raise salary by 10%

IF employee joins 15 years before 2020 then raise salary by 15%

Display the final salary, name and year.

1. Using the Account class as a base class, write two derived classes called Savings Account and Current Account. Account class should have attributes like account balance, loan (yes or no Boolean variable). A Savings Account object, in addition to the attributes of an Account object, should have an interest variable and a method which adds interest (assume your own interest) to the account if the account is associated with loan. A Current Account object, in addition to the attributes of an Account object, should have an overdraft limit variable to display if the account balance is less than 1000. Compute the interest and display the initial amount, loan amount, amount after deducting loan amount and check whether final balance is less than 1000 or not.
2. Create a class for opening a bank account with necessary details. Add two bank account. Create a subclass to delete account details only if the account number matches. Take the account number from user and match with already existing two account numbers. If matches display account removed else no account matched.
3. Write a program to extend the Vehicle class. Vehicle encapsulates information about vehicles, including the number of passengers they can carry, their fuel capacity, and fuel consumption rate. We can use the Vehicle class as a starting point from which more specialized classes are developed. For example, one type of vehicle is a truck. An important attribute of a truck is its cargo capacity. Thus, to create a Truck class, you can extend Vehicle, adding an instance variable that stores the carrying capacity. In the process, the instance variables in Vehicle will be made private, and accessor methods are provided to get and set their values. Particularly try to perform inheritance with private and protected access specifiers.
4. Create class SavingsAccount. Use a static variable annualInterestRate to store the annual interest rate for all account holders. Each object of the class contains a private instance variable savingsBalance indicating the amount the saver currently has on deposit. Provide method calculateMonthlyInterest to calculate the monthly interest by multiplying the savingsBalance by annualInterestRate divided by 12 this interest should be added to savingsBalance. Provide a static method modifyInterestRate that sets the annualInterestRate to a new value. Write a program to test class SavingsAccount. Instantiate two savingsAccount objects, saver1 and saver2, with balances of $2000.00 and $3000.00, respectively. Set annualInterestRate to 4%, then calculate the monthly interest and print the new balances for both savers. Then set the annualInterestRate to 5%, calculate the next month’s interest and print the new balances for both savers.
5. Create a class called Date that includes three pieces of information as instance variables—a month (typeint), a day (typeint) and a year (typeint). Your class should have a constructor that initializes the three instance variables. Create another called checkdate which inherits date. Check month should be less than equal to 12, year should be four digit number and less than 2020, day should be less than 31. Create another class daycheck which inherits checkdate. In this check whether the given day is in month or not (jan 31 days, feb either 28 or 29, march 31… and so on). Finally print whether the date given is correct or not.
6. Create a class called Employee that includes three pieces of information as instance variables—a first name (typeString), a last name (typeString) and a monthly salary (double). Your class should have a constructor that initializes the three instance variables. Provide a set and a get method for each instance variable. If the monthly salary is not positive, set it to 0.0. Write a test application named EmployeeTest that demonstrates class Employee’s capabilities. Create two Employee objects and display each object’s yearly salary. Then create another class which inherits employee to give each Employee a 10% raise and display each Employee’s yearly salary again.