Hands-on Assignments for Classes and Objects

No.	Hands-on Assignment	Topics Covered	Status
1	Create a class Box that uses a parameterized method to initialize the dimensions of a box.(dimensions are width, height, depth of double type). The class should have a method that can return volume. Obtain an object and print the corresponding volume in main() function.	Classes and Objects, Constructor	□
2	Create a new class called “Calculator” which contains the following: 1. A static method called powerInt(int num1,int num2) that accepts two integers and returns num1 to the power of num2 (num1 power num2). 2. A static method called powerDouble(double num1,int num2) that accepts one double and one integer and returns num1 to the power of num2 (num1 power num2). 3. Call your method from another class without instantiating the class (i.e. call it like Calculator.powerInt(12,10) since your methods are defined to be static) Hint: Use Math.pow(double,double) to calculate the power.	Classes and Objects, Constructor, static	□
3	Design a class that can be used by a health care professional to keep track of a patient’s vital statistics. Here’s what the class should do: 1. Construct a class called Patient 2. Store a String name for the patient 3. Store weight and height for patient as doubles 4. Construct a new patient using these values 5. Write a method called BMI which returns the patient’s BMI as a double. BMI can be calculated as BMI = (Weight in Pounds / (Height in inches x Height in inches)) x 703 6. Next, construct a class called “Patients” and create a main method. Create a Patient object and assign some height and weight to that object. Display the BMI of that patient.	Classes and Objects, Constructor, static	□

Encapsulation / Abstraction

No.	Hands-on Assignment	Topics Covered	Status
1	CCreate 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.	Encapsulation / Abstraction	□

Inheritance

No.	Hands-on Assignment	Topics Covered	Status
1	Create a class named ‘Animal’ which includes methods like eat() and sleep(). Create a child class of Animal named ‘Bird’ and override the parent class methods. Add a new method named fly(). Create an instance of Animal class and invoke the eat and sleep methods using this object.Create an instance of Bird class and invoke the eat, sleep and fly methods using this object.	Inheritance	□
2	Create a class called Person with a member variable name. Save it in a file called Person.java Create a class called Employee who will inherit the Person class.The other data members of the employee class are annual salary (double), the year the employee started to work, and the national insurance number which is a String.Save this in a file called Employee.java Your class should have a reasonable number of constructors and accessor methods. Write another class called TestEmployee, containing a main method to fully test your class definition.	Inheritance	□
3	A HighSchool application has two classes: the Person superclass and the Student subclass. Using inheritance, in this lab you will create two new classes, Teacher and CollegeStudent. A Teacher will be like Person but will have additional properties such as salary (the amount the teacher earns) and subject (e.g. “Computer Science”, “Chemistry”, “English”, “Other”). The CollegeStudent class will extend the Student class by adding a year (current level in college) and major (e.g. “Electrical Engineering”, “Communications”, “Undeclared”).	Inheritance	□

Overriding / Polymorphism

No.	Hands-on Assignment	Topics Covered	Status
1	Create a base class Fruit which has name ,taste and size as its attributes. A method called eat() is created which describes the name of the fruit and its taste. Inherit the same in 2 other class Apple and Orange and override the eat() method to represent each fruit taste.	Inheritance / Overriding	□
2	Write a program to create a class named shape. It should contain 2 methods- draw() and erase() which should print “Drawing Shape” and “Erasing Shape” respectively. For this class we have three sub classes- Circle, Triangle and Square and each class override the parent class functions- draw () and erase (). The draw() method should print “Drawing Circle”, “Drawing Triangle”, “Drawing Square” respectively. The erase() method should print “Erasing Circle”, “Erasing Triangle”, “Erasing Square” respectively. Create objects of Circle, Triangle and Square in the following way and observe the polymorphic nature of the class by calling draw() and erase() method using each object. Shape c=new Circle(); Shape t=new Triangle(); Shape s=new Square();	Polymorphism	□

String & String Buffer

No.	Hands-on Assignment	Topics Covered	Status
1	Write a Program that will check whether a given String is Palindrome or not	String/StringBuffer	□
2	Given two strings, append them together (known as "concatenation") and return the result. However, if the concatenation creates a double-char, then omit one of the chars. If the inputs are “Mark” and “Kate” then the ouput should be “markate”. (The output should be in lowercase)	String/StringBuffer	□
3	Given a string, return a new string made of n copies of the first 2 chars of the original string where n is the length of the string. The string may be any length. If there are fewer than 2 chars, use whatever is there. If input is "Wipro" then output should be "WiWiWiWiWi".	String/StringBuffer	□
4	Given a string of even length, return the first half. So the string "CatDog" yields "Cat". If the string length is odd number then return null.	String/StringBuffer	□
5	Given a string, return a version without the first and last char, so "Wipro" yields "ipr". The string length will be at least 2.	String/StringBuffer	□
6	Given 2 strings, a and b, return a string of the form short+long+short, with the shorter string on the outside and the longer string on the inside. The strings will not be the same length, but they may be empty (length 0). If input is "hi" and "hello", then output will be "hihellohi".	String/StringBuffer	□
7	Given a string, if the first or last chars are 'x', return the string without those 'x' chars, and otherwise return the string unchanged. If the input is "xHix", then output is "Hi".	String/StringBuffer	□
8	Given two strings, word and a separator, return a big string made of count occurrences of the word, separated by the separator string. if the inputs are "Wipro","X" and 3 then the output is "WiproXWiproXWipro".	String/StringBuffer	□
9	Return a version of the given string, where for every star (*) in the string the star and the chars immediately to its left and right are gone. So "ab*cd" yields "ad" and "ab**cd" also yields "ad".	String/StringBuffer	□
10	Given two strings, a and b, create a bigger string made of the first char of a, the first char of b, the second char of a, the second char of b, and so on. Any leftover chars go at the end of the result. If the inputs are "Hello" and "World", then the output is "HWeolrllod".	String/StringBuffer	□
11	Given two strings, a and b, create a bigger string made of the first char of a, the first char of b, the second char of a, the second char of b, and so on. Any leftover chars go at the end of the result. If input is “abc” and “xyz”, then output should be “axbycz”.	String/StringBuffer	□
12	Given a string and an int n, return a string made of n repetitions of the last n characters of the string. You may assume that n is between 0 and the length of the string, inclusive. For example if the inputs are “Wipro” and 3, then the output should be “propropro”.	String/StringBuffer	□
13	Given a string and a non-empty word string, return a string made of each char just before and just after every appearance of the word in the	String/StringBuffer	□

	string. Ignore cases where there is no char before or after the word, and a char may be included twice if it is between two words. If inputs are "abcXY123XYijk" and "XY", output should be "c13i". If inputs are "XY123XY" and "XY", output should be "13". If inputs are "XY1XY" and "XY", output should be "11".		