**Day 1 Assignments**

1. Identify the various classes, attributes, methods and the relationships between the classes for the below use case. Create appropriate class diagrams to depict the relationships.

* A company wants to build an application to store the details of their meeting rooms. There are two types of meeting rooms - one with Zoom calling facility and the second without Zoom calling facility.
  + All meeting rooms are identified using a unique identifier. The application should also maintain the information about the capacity of each meeting room and the floor in which it is located.
  + For meeting rooms with Zoom calling facility, the application must also maintain related information like Zoom device id and Zoom account id.
  + Employees can book meeting rooms by providing the following information - employee id, meeting date, meeting time, and duration of meeting.
  + All meeting bookings are also stored by the application and employees can get meeting information by providing the date of the meeting.

1. Write a method which accepts the name of a person as a parameter returns and returns a simple welcome message to the person. This method must be invoked by the main method and its output should be displayed. E.g. If the person enters “Naveen” then the method should return “Hello Naveen, Welcome to Java World! “
2. Write a method that accepts 3 numbers as parameters and returns the largest number among them.
3. Write a method which accepts a 3 digit number and prints it’s digits in words E.g. If the input number is 951, then the output should be “Nine Five One”
4. Implement the following methods in a class.

public void add (int a, int b)  
public void add (float a, float b)  
public void add (short a, short b)

Invoke these methods by passing appropriate parameters and display the output in the main method. Hint: To call add(short a, short b) method, you may need to use add((short)5,(short)6)

1. Implement appropriate versions of the add() method in a class so that

* add(10,20) returns 30
* add(10,20,30) returns 60
* add(10.5, 20.1) returns 30.6
* add(“Hello”,20) returns “Hello 20”

1. Define a class Student with the following attributes

* studentId of type integer
* studentName of type String
* city of type String
* marks1 of type integer
* marks2 of type integer
* marks3 of type integer
* feePerMonth of type float
* isEligibleForScholarship of type boolean

Implement the following methods in addition to the setter and getter methods for the various attributes

* getAnualFee() which returns the product of feePerMonth and 12
* getTotalmarks() which returns the sum of marks1, marks2 and marks3
* getAverage() which returns the average of marks1, marks2 and marks3
* getResult() which returns “pass” if the person has scored more than 60 in each subject, or returns “fail” otherwise

Create another class TestMain with the main() method which performs the following actions

* Creates three Student objects
* Populates the objects using the setter methods
* Displays the name of the Student who has the highest total marks
* Prints the name and fee of the Student who pays the least monthly fee
* Prints the name, total marks , average marks , result, and “Scholarship available” or “Scholarship not available” based on the student’s eligibility for every student.

Day 1 Assignments

1. Identify the classes, attributes, methods and the relationships between the classes for the below use case. You can show the relationship by using class diagrams.

* Xyz Hospital wants to store the details of their rooms, patients and doctors.
  + There are two types of patients In-patients and out-patients. Both the patients have some common properties like patient id, patient name, age, patient complaint.
  + For in-patients, the application should store additional details like date of joining, room no and date of discharge.
  + For out-patients, it should store additional details like date of visit, and doctor name.
  + Application also maintains details of all the doctors like doctor id, name, age, specialization.
  + Each room in the hospital is uniquely identified by room no, room name, capacity of the room and doctor in charge.

1. Write a method which accepts empno, name, age and country as parameters from the user and returns a string by adding all the details separated by “;”. Call this method from the main method and display the output. E.g. If the input is 12345, Naveen, 25 and India then method should return 12345;Naveen;25;India
2. Write a method that accepts 3 numbers as parameters and returns the sum of biggest and smallest numbers. E.g. if the input is 1 9 3, then method should return 10 (1+9) if the input is 2 9 2, then method should return 11(2+9). If the input is 9 9 9, then method should return 18(9+9)
3. Write a method which accepts a 3-digit number and prints the output as shown in the below example  
   E.g. If the input is 951, then the program should print “(9 x 100) + (5 x 10) +1”
4. Write the below methods in a class and provide the appropriate implementations.

public void add (long a, long b)

public void add (float a, float b)

public void add (short a, short b)

Call all the methods by passing different parameters.

Hint: to call method add(short a, short b) use add((short)5,(short)6)

1. Write necessary methods in a class based on the below rules

* Method call add(10,20) should return 30
* Method call add(10,20,30) should return 60
* Method call add(10.5, 20.1) should return 30.6
* Method call add(“Hello”,20) should return Hello 20

1. Write a class Product with the following attributes

* productId of type integer
* productName of type String
* description of type String
* price of type float
* maxDiscountAllowed of type float (it is in percentage)
* monthOfManufacture of type integer
* yearOfManufacture of type integer
* imported of type boolean (if true it means product is imported from other countries)  
  Write all the setter/getter for all the attributes along with the below methods
* getDiscountedPrice() it should return the price minus the discount amount
* getTotalPriceForItems(int count) it should return the price multiplied by number of items.
* getDiscountedPriceForItems(int count) it should return discounted price multiplied by number of items.  
  Create another class TestMain with the main method to perform the below actions
* Create three objects of the above Product class
* Populate the attributes of objects by calling setter methods
* Print the name of the Product with the highest price
* Print the name and price of the Product with highest discounted percentage
* For every Product, print the product name, price, and price after discount in a single line; If the value of the imported is true then print “Product is imported” otherwise print “Product is a local product”

## Day2 Assignments

Note: For the following assignments create an exclusive class called Tester which contains the main method. Create objects of other classes, make calls to the methods, and test your code using this Class’s main method.

1. Write a method that accepts two integers - n1 & n2 and prints the sum of all the factors for each integer between n1 and n2. E.g. For n1=4 and n2=6,

* Factors for 4 are 1, 2 & 4. Sum of factors is 1+2+4 =7
* Factors for 5 are 1&5. Sum of factors is 1+5=6
* Factors for 6 are 1, 2, 3 & 6. Sum of factors is 1+2+3+6=12  
  Output: 7, 6, 12

1. Write a method which accepts a number n as a parameter and prints n0 to n10 as shown below using for loop, while loop and do while loop. Hint: Use Math.pow(x,y) method which returns xy in Java.  
   E.g. For parameter 2, it should print  
   2 power 0 = 1  
   2 power 1 = 2  
   ...  
   2 power 10 = 1024
2. Write a method that accepts a 5-digit number N and returns the Category Code of the number. The category code is generated as follows

* Step 1: Convert the number to a single digit number by adding all the numbers
* Step 2: if the result is not a single digit number repeat step 1
* Step 3: assign the code based on the below rules
  + For number 1 to 5 --> Category code is “First”
  + For number 6 or 9 --> Category code is “Second”  
    E.g. For input 12345
* Step 1: 1+2+3+4+5 =15
* Since 15 is a two-digit number repeat step 1; 1+5=6
* As per step 3, Category code is “Second”

1. Write a method which accepts a string as a parameter and returns the number of words in the String.
2. Write a program to test the below methods of String class   
   a. charAt   
   b. contains   
   c. length   
   d. indexOf   
   e. equals   
   f. equalsIgnoreCase   
   g. join   
   h. lastIndexOf   
   i. substring   
   j. tolowercase   
   k. touppercase l. trim
3. Write a class ArrayStore which contains one instance variable of type integer array. Write the necessary methods for the following operations. a. Accept 10 numbers and store these numbers into the array. b. Displays all the numbers using while & for loops c. Sort the array d. Accepts a number and returns the number of occurrences of the accepted number in the array e. Insert a number at a specific position f. Return an array by removing duplicates. E.g. if the elements of the array are [9,2,2,9,10,9] then method should return [9,2,10]
4. Write a method which accepts a two-dimensional array and returns the sum of all the numbers in the diagonal  
   E.g. Method should return 15 (1+5+9) for the below array  
   [1 2 3  
   4 5 6  
   7 8 9 ]
5. Write a method which accepts array of size 10. The method should convert it into a variable sized two-dimensional integer array as illustrated in the example and return it. E.g. Consider the input array containing [1, 3, 2, 4, 5,9,8,6,7,10], Return the array of below numbers.  
   1  
   3 2  
   4 5 9  
   8 6 7 10

**Day 2 Assignments**

Note: For the following assignments create an exclusive class called Tester which contains the main method. Create objects of other classes, make calls to the methods, and test your code using this Class’s main method.

1. Write a method which accepts two integers - n1 & n2 and displays the prime numbers between n1 andn2  
   E.g. if n1 = 5 and n2 = 10, then the method should display 5 and 7
2. Write a method that accepts an integer as parameter and displays its multiplication table using for loop, as shown below. Implement 2 other methods which perform the same action but using a while and do-while loop respectively. E.g. if the input parameter is 2, then these methods must output the multiplication table as follows:  
   2 x 1 = 2  
   2 x 2 = 4  
   ...  
   2 x 10 = 20
3. Write a method that accepts a 5 digit number N as parameter and returns the largest number possible by rearranging its digits.  
   E.g. If N is 41459 then the method should output 95441
4. Write a method which accepts a string as parameter and returns the number of words in it.
5. Create a program with methods to test the functionality of the various methods of the String class

* charAt
* contains
* length
* indexOf
* equals
* equalsIgnoreCase
* join
* lastIndexOf
* substring
* tolowercase
* touppercase
* trim

1. Define a class ArrayStore which contains an integer array as its instance variable. Create necessary methods for the following operations.

* Accept 10 integers and store them into the array.
* Display the elements of the array using while & for loops
* Sort the array
* Accept a number and return the number of times it occurs in the array
* Insert a number into the array at a specified position
* Return array that excludes duplicate elements in the original array. E.g. if the elements of the original array are [9,2,2,9,10,9] then return [9,2,10]

1. Write a method that accepts a two-dimensional array and returns the sum of all the elements in it.  
   E.g. The method should return 12(1+2+1+2+2+2+1+0+1) for the below array  
   [ 1 2 1  
   2 2 2  
   1 0 1 ]
2. Write a method which accepts an array of Strings. The method should convert the input into a variable sized two-dimensional character array and return it.  
   E.g. For an input of the following 3 strings, “Hello” “Hi” “Welcome”. The method should return the below array of characters.  
   'H' 'e' 'l' 'l' 'o'  
   'H' 'i'  
   'W' 'e' 'l' 'c' 'o' 'm' 'e'

## Day 3 Assignments

Note: All classes should be organized into packages that follow a well-defined hierarchy.

1. A Job tracker application is used for tracking the activities of a team. Each activity is represented as a Job class. Create a class Job with the attributes job name, owner, effort required (in months), month of creation, year of creation, status (not started/ work in progress/ completed). Write necessary methods to accept and display the information. Create the constructors based on the below rules.

* Job name and owner are mandatory fields and should be supplied at the time of creating a class
* Compiler should raise an error when we try to create Job object without passing any parameters.
* Write a constructor which accepts all the attributes as parameters while creating the object. From this constructor call the constructor (mentioned at point a) to initialize the mandatory fields

1. In the above class create a static variable jobsCount. Write necessary methods to get its value. Every time an object of Job is created, increment the value of jobsCount variable by 1.
2. Create a jobId field. Make this variable as a readOnly (make it private and write only getter method). Generate jobId value by using the below formula  
   jobId= "jobName" + "\_" + jobsCount  
   E.g. if the Job name is “Maintain Server” and the value of jobsCount variable is 31, then store "Maintain Server\_31" in jobId
3. Class PriorityJob contains all the attributes of Job class and other attributes to store the priority of the job (low, medium, high) and monitoredBy field to store the name of the person monitoring the Job. Another class MultiOwnerJob contains all attributes of Job class in addition to the name of the second owner.

* Write necessary classes, constructors and methods for storing and displaying additional information. Hint: use super keyword to call the methods/constructor of super class.

1. In the Job class write a method called showDetails() which returns a String by concatenating all the field values.
2. Override showDetails() method of Job (created in Day 3) in its the sub classes to include the details of additional fields. Use super key word to call the methods of super class if required.
3. Create an array of Job class and store Objects of Job, PriorityJob, and MultiOwnerJob in the array.
4. Using a single for loop try traverse the above array and call the method showDetails() on all the objects of the array. Understand the concept of runtime polymorphism.
5. Check whether it is possible to call all the methods of PriorityJob and MultiOwnerJob while traversing the array. If not use typecasting to achieve the above task.

**Day3 Assignments**

Note: All classes should be organized into packages that follow a well-defined hierarchy.

1. Create a class Movie with attributes - movie name, produced by, directed by, duration, year, category (comedy/action/..). Write necessary methods that accept and display the information. Create the constructors based on the below rules.

* Movie name and produced by are mandatory fields and should be supplied at the time of creating the object
* Compiler should raise an error when you try to create Movie object without passing any parameters.
* Write a constructor which accepts all the attributes as parameters while creating the object. From this constructor call the constructor (mentioned at point a) to initialize mandatory fields

1. In the above class create a static variable moviesCount. Write necessary methods to get the values. Every time an object of Movie is created, increment the value of moviesCount variable.
2. Create a movieId field. Make this variable as a readOnly variable (i.e. make it private and write only a getter method). Generate movieId value by using the below formula  
   movieId=”movieName”+”\_”+moviesCount  
   eg. if the Movie name is “Hello” and the value of moviesCount variable is 31, then store Hello\_31
3. Define a new class SpecialMovie which contains all the attributes of Movie and other attributes to store the technology used for soundEffects and visualEffects. Define another class InternationalMovie which contains all attributes of Movie class and other attributes to store country and language.

* Write necessary classes, constructors and methods for storing and displaying additional information. Hint: use super keyword to call the methods/constructor of super class.

1. In the Movie class write a method called showDetails() which concatenates the value of all the attributes and returns it as a String