# Core java -Assignment List

## OOPS Concepts

- Program demonstrating basics of all OOPS concepts like encapsulation, polymorphism and inheritance.

## Java programming fundamentals

(Source Code File declaration rules, classes / objects, inner class, constructors, access and non-access modifiers, primitive data types, instance variables, enums, String Class)

1. Guessing game which involves a game object and 3 player objects. The game generates a random number between 0 and 9 and the 3 player objects try to guess it.
2. Create your own calendar class and perform various operations on it with
3. Create a box class having instance variables width and height and having various instance methods with and without arguments and performing various operations on it. It must have default and parameterized constructors too. Use this variable in the constructors and perform garbage collection using finalize() method.
4. Stack operations using class, objects, instance variables, constructors, garbage collector and super keyword.
5. Create class Student (roll number, name, number of subjects, marks of each subject).No. of subjects varies for each student. Write a parameterized constructor which initializes roll number, name, number of subjects and create the array of marks dynamically. Display details of all students with percentage and class obtained. Use inner class too.
6. Using class implement HASHTable to:
   - ✓ Accept records of n students(name, percentage).
   - ✓ Display details of all students. Find out highest marks.

7. Demonstration of Java Program with logical operators, shorthand assignment operators, condition operators, bitwise operators.
8. Java program with use of type casting.
9. Accept the number from command line and calculate sum of digits.
10. Java program to illustrate the various scopes of variables: static scope, block scope, method local scope, instance scope.
11. Demonstrate how one class can be defined in another class.
12. Copying of one array into another, sort list of numbers.
13. Program to illustrate concatenation of 2 strings and usage of various string methods and alphabetic ordering of strings.
14. Demonstration of comparison between equals() and ==.
15. Demonstrate manipulation of Strings using StringBuffer and StringBuilder classes.
16. Demonstration of replace() method from StringBuffer which is used to replace full string.
17. Take 2 String inputs from user. Convert String 1 in upper and String 2 in lowercase. Concatenate both strings and display results.
18. Pascal Triangle using array.
19. Java Program that inputs a person's name in form of First Middle Last, and then prints it in form Last First M., where "M" is person's middle initial.
20. Perform operations on matrix.
21. Accept 2 strings as command line arguments. Check if 2nd string is substring of 1st string.
22. Convert string into decimal, binary and hexadecimal.
23. Demonstration of randomly automated decision maker using Random Class.
24. Find square root of number from Math Class.

# *Package, Inheritance, Method overriding/overloading, interfaces, abstract class, static modifier*

1. Box class using overloaded constructors and objects as parameters and return values and recursion.
2. Program demonstrating static modifiers and instance and static init blocks.
3. Bank Program:
   Customer having savings account. Design base class Customer(name, phone number).Derive a class depositor(accno, balance) from customer. Derive class Borrower(loan-no, loan-amt) from Depositor. Write necessary member functions to read and display details of n customers using method overriding also.
4. Create a class telephone containing name, telephone number and city and write member functions for the following:
   - ✓ Search telephone number with given name.
   - ✓ Search name with given telephone number
   - ✓ Search all customers in given city(Using Function Overloading)
5. Define an Employee class with suitable attributes having getSalary() method, which returns salary withdrawn by particular employee. Write class Manager which extends class Employee, override the getSalary() method, which will return salary of manager by adding travelling allowance, house rent allowance etc.
6. Create an abstract class Person. Define 2 classes Employee and Worker from it. Use Proper method to accept and display the details for the same. The fields of Employee are Emp_no, Emp_name, address. Similar fields for worker are name and working hours.
7. Write Java Program to accept "n" numbers through the command line and store all the prime numbers and perfect numbers into the different arrays and display both the arrays.
8. Class Shape consists of one final method area() and volume().Create 3 subclasses Rectangle, Circle and Triangle and calculate area and volume of it.
9. Create an interface Shape. Derive 3 classes sphere, cone and cylinder from it. Calculate area and volume of all(using method overriding).
10. Write package for Games, which have 2 classes Indoor and Outdoor. Use a function display() to generate the list of players for the specific games. Use default and parameterized constructors. Make use of protected access specifier.

11. Create a package TYBSc which will have 2 classes as class Mathematics with methods to add 2 numbers and 3 float numbers and class Maximum with a method to find the maximum of 3 nos.
12. Accept name, address, rollno, percentage with base class and cast annual income in derived class and check if scholarship is sanctioned or not. If cast is not open and annual income is less than 100000 then scholarship is sanctioned.
13. Create base class college(name, address) and generate from college one derived class principal(name, address).Derive a class from Principal i.e. Salary(sal) and print all the attributes.
14. Write a package StrPack having 2 classes Con and Comp. Con Class has to concatenate 2 strings and Comp Class compare 2 strings. Display the proper message on execution.
15. Create package vehicle which will have 2 classes as class 2-wheeler and 4-wheeler. 2-wheeler with method(CC, price). 4-wheeler with method show(regno, regyear).

## *Exception Handling and Assertion*

1. Write Java Program to accept a number from user, if number is zero throw user defined exception "Number is 0" otherwise check whether no. is prime or not(use static keyword).
2. Write class Student with attributes roll no, name, age and course. Initialize values through parameterized constructor. If age of student is not in between 15 and 21 then generate user defined exception "Age Not within Range". If name contains numbers or special symbols, raise exception "Name not Valid".
3. Write a Java Program to find the exception Marks Out of Bounds. Create a class Student. If mark is greater than 100, it must generate user defined exception called Mark Out of Bounds Exception and throw it.
4. Find out maximum of array elements and check for array limit.
5. Write a class Account with acc_no, name and balance. Initialize values through parameterized constructor. If balance is between 1000 and 5000, generate user defined exception "Balance within the range". If name contains digits, raise exception "Name not Valid".

6. Class ExceptionDemo throws following exception depending upon following condition:
   - ✓ Take any integer from the keyboard.
   - ✓ If the integer is between 0 and 5, exception of type "Small Number" is thrown.
   - ✓ If the number is between 5 and 10,"Proper Number" is thrown.
   - ✓ If the number is greater than 10,"Greater Number" is thrown.
   - ✓ Also find the factorial of that number(using static keyword).
7. Class ExceptionDemo throws the following exception depending upon the following conditions. Take any string from the keyboard.
   - ✓ If string length is between 0 and 5, then Exception "too small string" is thrown.
   - ✓ If string length is between 5 and 10, then exception to type "Perfect String" is thrown.
8. Program which accepts string from user. If string contains word "India", raise an exception.
9. Write a Java program to find sum of integer from command line argument and count invalid integers through command line(Use Exception Handling).
10. Write a program in Java to check if entered data is alphabet or digit. If it is alphabet, then print if it is capital or small case. If digit is entered throw user defined exception "digit not allowed".
11. Write program demonstrating nested try blocks and multiple catch statements and using throw and throws and finally keywords.

## *File Handling and Serialization*

1. Write Java program to accept list of files as command line arguments. Display the name and size of all the files. Delete all files with extension as .html from current directory. Appropriate error messages should be printed.
2. Write program in Java to read, update and delete any record from database. The database will be created in appropriate backend.
   - ✓ Database about element(oxygen, hydrogen) has following fields(Atomic weight, Name, Chemical Symbol).
   - ✓ The input should be provided through command line argument.
   - ✓ It will read and show the contents of the table.
   - ✓ Will update record according to the name specified.

- ✓ Use file handling concept.

3. Display number of characters, lines, words from a file.
4. Create 2 files F1 and F2. Copy contents of file F1 by changing the case into file F2.

   F2=F2.

   Also copy the contents of F1 and F2 in F3.

   F3=F1+F2.

   Display contents of F3 (using command line arguments).
5. Using Class implement HashTable to
   - ✓ Accept n records of students(Name, Percentage)
   - ✓ Display details of all students. Find out highest marks.
6. Copying characters from 1 file(input.dat) to another file(output.dat).
7. Write bytes to file using FileOutputStream class.
8. Use DataInputStreams for reading textual input line by line with an appropriate BufferedReader.
9. Count Number of lines in file using FileReader class.
10. Programme to determine console input and output.
11. Handle College database and Factory ERP database using serialization.
12. Demonstration of random access file.
13. Input a persons name in form of First Middle Last, and then print it in form Last First Middle, where "M" is persons middle initial.
14. Accept a file name from user as command line argument and display every alternate character of file. Generate exceptions for insufficient no. of arguments and file not found.
15. Write Java Program which accepts n integers from user and writes only the even numbers to file "even.txt".

## Multithreading

1. Thread functionalities like getName, setName, getPriority, setPriority, sleep, yield, join etc. by creating threads using Thread Class and Runnable interface.
2. Shared Bank Account using Synchronization and object locks and wait and notify methods.
3. Write Time display Applet which prints date and time on screen.
4. Inter-thread communication using wait, notify, notifyAll methods.
5. Program in JAVA showing lifecycle of thread. Print randomly the name of the thread and value of sleep time. The name of thread should be hardcoded through constructor. Sleep time should be random integer in range 0 to 4999.
6. Display name and priority of current thread. Change name of Thread to MyThread and set priority to 2 and display it on screen.
7. Program to suspend and resume threads.
8. Demonstrate synchronized block.

## Collection Framework

1. Demonstrate ArrayList of Strings and sort them. Demonstrate ArrayList of Library Class Objects using generics and sort them appropriately.
2. Demonstrate the hasNext, next() and remove() methods of Iterator interface.
3. Write Java Program to read n names of friends, store into linked list and display content of same list.
4. Write program to accept name and roll number of student and store it in table and display contents of hash table.
5. Demonstration of Iterator, Comparator and Enumeration Interface.
6. Write program to accept names of n cities, insert the same into Arraylist collection and display the content of the same array list and remove these elements.
7. Program to read n elements from user, insert into stack and remove these to display.
8. Program to create linked list of integer objects and do following operations:
   - ✓ Add element to 1st position.
   - ✓ Delete last element.
9. Program demonstrating LinkedHashMap and TreeMap.
10. Usage of ArrayList with command line arguments.
11. Demonstrate usage of Vectors.
12. Read in a series of 1st names and store them in LinkedList. Program should not allow duplicate names and it should allow user to search for 1st name.

## Wrapper Class

1. Demonstration of the xxxValue(), valueOf() and parseXxx() and toString() methods of class Integer.
2. Demonstration of various method overloading rules of wrapper classes.
3. Demonstration of equals() and == operator between wrapper objects.