**JAVA**

**Introduction to Java**

**Lab Exercise No:** 1

**Exercise Objective(s):** *Simple java program*

**Exercise:** *Write a program with a class name “Welcome” and display a message as follows: “Welcome*

*to the world of Java”*

**Recommended duration:** *10Mins*

**Solution Guidance (if applicable):** *NA*

**Lab Exercise No:** 2

**Exercise Objective(s):** *Compilation and execution from command line, Concept of object and class*

**Exercise:** *Write a program that takes a console input (Input given by the user while executing the*

*program in command line) and prints the same.*

**Recommended duration:** *10Mins*

**Solution Guidance (if applicable):** *The input can be printed as follows.*

*System.out.println ("Message: "+args [0]);*

**Lab Exercise No:** 3

**Exercise Objective(s):** *Comments in java programs and java documentation*

**Exercise:** *Write a program with all the type of comments and execute it. User nested comments also.*

**Recommended duration:** *10Mins*

**Solution Guidance (if applicable):** *NA*

**JAVA**

**Basic elements of Java**

**Lab Exercise No:** 4

**Exercise Objective(s):** *Primitive data types and their range, Variables, Constants and literals,*

*Conventions*

**Exercise:** *Write a program which declares variables of int, float, double data types and a constant of*

*long data type and displays all with an appropriate message. Follow the naming conventions*

*for all the variables and literals.*

**Recommended duration:** *10Mins*

**Solution Guidance (if applicable):** *NA*

**Lab Exercise No:** 5

**Exercise Objective(s):** *Simple operators*

**Exercise:** *Write a program to get two numbers as input through command line and swap the values of*

*two numbers without using a temporary variable and display the same.*

**Recommended duration:** *20Mins*

**Solution Guidance (if applicable):** *a = a + b*

*b = a – b*

*b = a - b*

**Lab Exercise No:** 6

**Exercise Objective(s):** *Conditional statements*

**Exercise:** *Write a program to determine whether the given year is leap year or not(Get the input*

*through command line).*

**Recommended duration:** *20Mins*

**Solution Guidance (if applicable):** *NA*

**Lab Exercise No:** 7

**Exercise Objective(s):** *Conditional statements*

**Exercise:** *Write a program to determine the largest of three numbers.*

**Recommended duration:** *20Mins*

**Solution Guidance (if applicable):** *NA*

**Lab Exercise No:** 8

**Exercise Objective(s):** *Loops*

**Exercise:** *Write a program to determine whether a number is a palindrome or not.*

**Recommended duration:** *20Mins*

**Solution Guidance (if applicable):** *NA*

**Lab Exercise No:** 9

**Exercise Objective(s):** *Loops*

**Exercise:** *Write a program to display the Fibonacci series starting from 0 till 200.*

**Recommended duration:** *20Mins*

**Solution Guidance (if applicable):** *NA*

**Lab Exercise No:** 10

**Exercise Objective(s):** *Constants and literals, Loops*

**Exercise:** *Write a program to declare a set of 5 words and reverse each word and arrange the resulting*

*words in alphabetical order and display the same.*

**Recommended duration:** *20Mins*

**Solution Guidance (if applicable):** *NA*

**Lab Exercise No:** 11

**Exercise Objective(s):** *Constants and literals, Loops*

**Exercise:** *Write a program to arrange an array of elements in ascending order using selection sort*

*algorithm.*

**Recommended duration:** *20Mins*

**Solution Guidance (if applicable):** *NA*

**Lab Exercise No:** 12

**Exercise Objective(s):** *Conditional statements, Loops*

**Exercise:** *A shopkeeper sells three products whose retail prices are as follows:*

*Product 1 - 22.50*

*Product 2 - 44.50*

*Product 3 - 9.98*

*Write an application that reads a series of pairs of numbers as follows:*

*a) Product number*

*b) Quantity sold*

*The application should use a switch statement to determine the retail price for each product. It*

*should calculate and display the total retail value of all products sold.*

**Recommended duration:** *20Mins*

**Solution Guidance (if applicable):** *NA*

**Lab Exercise No:** 13

**Exercise Objective(s):** *Simple operators, Conditional statements, Loops*

**Exercise:** *Consider user has N eggs. Then display the no of eggs in gross (144 eggs make one gross) and*

*no of eggs in dozen (12 eggs make one dozen) and the no of eggs that is left out remaining.*

*The total no of eggs can be got as input through command line. The program should display*

*how many gross, how many dozen, and how many left over eggs the user has.*

**Recommended duration:** *20Mins*

**Solution Guidance (if applicable):** *For example, if the input is 1342 eggs, then the program should respond with*

*Your number of eggs is 9 gross, 3 dozen, and 10*

**JAVA**

**Classes and Methods**

**Lab Exercise No:**14

**Exercise Objective(s):***Construction of an object*

**Exercise:***Create a class called Calculator which has 4 different methods add, diff, mul and div which*

*accepts two numbers as parameters. Create an object to access these methods and invoke*

*these methods with two numbers and display the result in the corresponding methods.*

**Recommended duration:***20Mins*

**Solution Guidance (if applicable):** *NA*

**Lab Exercise No:**15

**Exercise Objective(s):***Construction of an object, constructors*

**Exercise:***Create a class called Sample. Write a program to display the no of objects created for that*

*class or the no of times that class is instantiated.*

**Recommended duration:***15Mins*

**Solution Guidance (if applicable):** *Use a static int variable and increment that variable inside the*

*constructor.*

**Lab Exercise No:**16

**Exercise Objective(s):***Construction of an object, this keyword, accessors(getters) and mutators(setters),*

*public and private access specifiers, instance and class member variables*

**Exercise:***Create a class called Student with the following details: RollNo, StudName, MarksInEng,*

*MarksInMaths and MarksInScience. Write getters and setters for the all variables. RollNo*

*should be automatically generatedwhenever a newstudent is added.*

*Create a class called Standard with 8 students’ details and write separate method for each of*

*the following tasks and invoke the same.*

1. *To display the entire roll no and the name of the students in the class in the ascending order of roll no.*
2. *To display the roll no and the name of the student who has got the highest percentage.*
3. *To display the roll no and the name of the student who scored highest mark*

*inmathematics.*

1. *To display the roll no and the name of the student in the ascending order of the total marks in mathematics and science alone.*
2. *To display the roll no, name, total marks, percentage and rank of all the students in the descending order of rank.*

**Recommended duration:***40Mins*

**Solution Guidance (if applicable):** *NA*

**Lab Exercise No:**17

**Exercise Objective(s):***String class, String immutability*

**Exercise:***Write class that declares the following String.*

***“The quick brown fox jumps over the lazy dog”.***

*Perform the following modifications to the above string using appropriate methods.*

1. *Print the character at the 12th index.*
2. *Check whether the String contains the word “is”.*
3. *Add the string “and killed it” to the existing string.*
4. *Check whether the String ends with the word “dogs”.*
5. *Check whether the String is equal to “The quick brown Fox jumps over the lazy Dog”.*
6. *Check whether the String is equal to “*THE QUICK BROWN FOX JUMPS OVER THE LAZY DOG*”.*
7. *Find the index position of the character “a”.*
8. *Find the last index position of the character “e”.*
9. *Find the length of the String.*
10. *Check whether the String matches to “The quick brown Fox jumps over the lazy Dog”.*
11. *Replace the word “The” with the word “A”.*
12. *Split the above string into two such that two animal names do not come together.*
13. *Print the animal names alone separately from the above string.*
14. *Print the above string in completely lower case.*
15. *Print the above string in completely upper case.*

**Recommended duration:***30Mins*

**Solution Guidance (if applicable):** *NA*

**Lab Exercise No:**18

**Exercise Objective(s):***String class, String immutability*

**Exercise:***Write a program to demonstrate the difference between equals and == operator with*

*appropriate example.*

**Recommended duration:***15Mins*

**Solution Guidance (if applicable):** *NA*

**Lab Exercise No:**19

**Exercise Objective(s):***Arrays*

**Exercise:***Write a program to declare an array with 8 elements and copy the 8 elements into another*

*array and display the same.*

**Recommended duration:***15Mins*

**Solution Guidance (if applicable):** *NA*

**Lab Exercise No:**20

**Exercise Objective(s):***Arrays*

**Exercise:***Write a program to display the sum and the average of elements in the array.*

**Recommended duration:***15Mins*

**Solution Guidance (if applicable):** *NA*

**Lab Exercise No:**21

**Exercise Objective(s):***Arrays*

**Exercise:***Write a program to construct two matrices and display the sum of those.*

**Recommended duration:***20Mins*

**Solution Guidance (if applicable):** *NA*

**Lab Exercise No:**22

**Exercise Objective(s):***Arrays*

**Exercise:***Write a program to display the square of the elements of a two dimensional array.*

**Recommended duration:***20Mins*

**Solution Guidance (if applicable):** *NA*

**Lab Exercise No:**23

**Exercise Objective(s):***Arrays*

**Exercise:***Write a program to construct an array with 10 elements and to find the number of*

*occurrences of each element in the Array.*

**Recommended duration:***20Mins*

**Solution Guidance (if applicable):** *NA*

**Lab Exercise No:**24

**Exercise Objective(s):***Overloading*

**Exercise:***Create a class called shape with the following methods*

1. *area*
2. *perimeter*

*Overload the area and perimeter method to calculate for both square and rectangle.*

*Create a main class and invoke the area method to calculate the area of the square and*

*rectangle. Also invoke the perimeter method to calculate the perimeter of the square*

*and rectangle.*

**Recommended duration:***20Mins*

**Solution Guidance (if applicable):** *NA*

**Lab Exercise No:**25

**Exercise Objective(s):***Overloading*

**Exercise:** *Create a class called employee with the following data members*

1. *empName*
2. *empId*
3. *empAge*
4. *empdesgn*
5. *empLocation*
6. *empExpInYrs*

*All these data members should be initialized using constructors. Use constructor overloading*

*and demonstrate by creating different employee objects with*

1. *Employee name alone*
2. *Employee name and id*
3. *Employee name, id and age*
4. *Employee name, id and designation*
5. *Employee name, id, age and designation*
6. *Employee name, id, age and location*
7. *Employee name, id, age and experience*
8. *Employee name, id, designation and experience*
9. *Employee name, id, designation, location and experience*
10. *Employee name, id, age, designation, location and experience*

**Recommended duration:***20Mins*

**Solution Guidance (if applicable):** *NA*

**Lab Exercise No:**26

**Exercise Objective(s):***Overloading*

**Exercise:***Create a class called Calculator which has 4 different methods add, diff, mul and div which*

*accepts two numbers as parameters. Overload the methods such that the parameters can be*

*of the following pattern.*

1. *Both are of int data type.*
2. *Both are of double data type.*
3. *First parameter is of int data type and second parameter is of double data type.*
4. *First parameter is of double data type and second parameter is of int data type.*

*Create anobject to access these methods and invoke these methods with different type of*

*numbers and display the result in the corresponding methods.*

**Recommended duration:***20Mins*

**Solution Guidance (if applicable):** *Re-use the code from Lab Exercise 14*

**Lab Exercise No:**27

**Exercise Objective(s):***Initializers*

**Exercise:***Write a class called Computer such that the object of that class should be created only when*

*the class is loaded.*

**Recommended duration:***20Mins*

**Solution Guidance (if applicable):** *NA*

**Lab Exercise No:**28

**Exercise Objective(s):***Var-args*

**Exercise:***In the calculator (Lab exercise - 14) program, make the add and diff method to accept var-args*

*and demonstrate.*

**Recommended duration:***20Mins*

**Solution Guidance (if applicable):***NA*

**JAVA**

**Packages**

**Lab Exercise No:**29

**Exercise Objective(s):***Package*

**Exercise:***Create a package called shapes. Create some classes in the package representing some*

*common geometric shapes like Square, Triangle, Circle and so on. Create a class called*

*TestShapes and create objects for all the shapes and print corresponding messages.*

*Execute the TestShapes class.*

**Recommended duration:** *20 Mins*

**Solution Guidance (if applicable):** *NA*

**Lab Exercise No:**30

**Exercise Objective(s):***Jar*

**Exercise:**

1. *Create a new project in which create a package named org.animals. In that create various classes like Lion, Tiger, Deer, Monkey, Elephant and Giraffe. In each class create data members like color, weight,age etc. Create methods like isVegetarian, canClimb, sound etc*
2. *Create another project and in that create a package called zoo and create a class called DelhiZoo and create objects for the animals that are existing in zoo and print the characteristic of each animal.*

**Recommended duration:***15Mins*

**Solution Guidance (if applicable):***Export the jar and add it as an External Archive.*

**Lab Exercise No:**31

**Exercise Objective(s):***System class*

**Exercise:***Create a class which displays the following about the JVM.*

1. *Version of Java*
2. *Vendor for Java*
3. *Class Path*
4. *Installed home directory*
5. *OS name on which it is installed with version*

**Recommended duration:***10Mins*

**Solution Guidance (if applicable):** *NA*

**Lab Exercise No:**32

**Exercise Objective(s):***Scanner class*

**Exercise:***Create a class called Student. Get the details like name, degree, age, total marks and*

*percentage from the user and display the same.*

**Recommended duration:** *20 Mins*

**Solution Guidance (if applicable):** *NA*

**Lab Exercise No:**33

**Exercise Objective(s):***Systemclass,usingstaticimport*

**Exercise:***Create a Package called house. Create 2 classes namely Hall and Kitchen.*

1. *In the Hall class print the message “This is the first room while entering the house” without using the class name System explicitly in the println statement.*
2. *In the Kitchen class create an array called appliances and initialize with values and print the same.*
3. *After printing copy that array into a different array.*
4. *Invoke garbage collector explicitly for the Kitchen class.*

**Recommended duration:***20Mins*

**Solution Guidance (if applicable):** *NA*

**JAVA**

**Inheritance**

**Lab Exercise No:**34

**Exercise Objective(s):***The concept of inheritance*

**Exercise:***Create a class called Vehicle. Create subclasses like Truck, Bus, Car etc. Add common methods*

*in the base class and specific methods in the corresponding class. Create a class called Road*

*and create objects for the Truck, Car, Bus etc and display the appropriate message.*

**Recommended duration:** *20 Mins*

**Solution Guidance (if applicable):** *NA*

**Lab Exercise No:**35

**Exercise Objective(s):***super keyword*

**Exercise:***In the Lab Exercise 34, in the Vehicle class constructor initialize few variables like color, no of*

*wheels, model etc. Give appropriate values for these variables from the invoking subclass.*

**Recommended duration:** *20 Mins*

**Solution Guidance (if applicable):** *NA*

**Lab Exercise No:**36

**Exercise Objective(s):***protected access specifier*

**Exercise:***In the Lab Exercise 35, create another class called City which creates an object for the Car,*

*Truck and Bus class and displays the details through a display () method in the Vehicle class.*

*The other methods and data members should not be accessible by the City class.*

**Recommended duration:** *20 Mins*

**Solution Guidance (if applicable):** *NA*

**Lab Exercise No:**37

**Exercise Objective(s):***Overriding*

**Exercise:** *In the Lab Exercise 30, create a super class called Animal and make all the existing classes as*

*the sub class for Animal class. Move the method isVegetarian and canClimb to the super class*

*and implement generically. Whenever necessary change the implementation of these methods*

*in the respective subclasses.Display the characteristic of each animal.*

**Recommended duration:** *20 Mins*

**Solution Guidance (if applicable):** *NA*

**Lab Exercise No:**38

**Exercise Objective(s):***final Keyword*

**Exercise:***In the Lab Exercise 37, make the Lion, Tiger, Deer, Monkey, Elephant and Giraffe classes such*

*that these classes cannot be inherited.*

**Recommended duration:** *20 Mins*

**Solution Guidance (if applicable):** *NA*

**Lab Exercise No:**39

**Exercise Objective(s):***Polymorphism*

**Exercise:***Create a class called Worker. Write classes DailyWorker and SalariedWorker that inherit from*

*Worker.Every worker has a name and a salaryrate. Write method Pay (int hours) to compute*

*the week pay of every worker. A Daily worker is paid on the basis of the number of days*

*she/he works.The salaried worker gets paid the wage for 40 hours a week no matter what the*

*actual hours are. Test this program to calculate the pay of workers.*

**Recommended duration:** *20 Mins*

**Solution Guidance (if applicable):** *NA*

**Lab Exercise No:**40

**Exercise Objective(s):***Polymorphism*

**Exercise:***ModifyLab Exercise 38 by creating intermediate Super classes called WildAnimals and*

*DomesticAnimals and create corresponding subclasses for the same. Create generic*

*methods in the super class and display the details of wild animals, domestic animals in*

*general separately and also display the details of each animal separately.*

**Recommended duration:** *20 Mins*

**Solution Guidance (if applicable):** *NA*

**Lab Exercise No:**41

**Exercise Objective(s):***abstract classes*

**Exercise:***Create a class called Shape3D with the following method signatures alone, volume () and*

*surfaceArea (). Then create subclasses like Cylinder, Sphere, and Cube etc and implement*

*these methods.*

**Recommended duration:** *20 Mins*

**Solution Guidance (if applicable):** *NA*

**Lab Exercise No:**42

**Exercise Objective(s):***abstract classes*

**Exercise:***Create the classes required to store data regarding different types of courses that employees*

*Ina company can enroll for. All courses have name and course fee. Courses are also either*

*classroom delivered or delivered online. Courses could also be full time or part time. The*

*program must be menu based input which enables the course coordinator to register*

*employees for courses, list out employees registered for specific courses, deregister employees*

*from a course.*

**Recommended duration:***30Mins*

**Solution Guidance (if applicable):***The output can be like this,*

*Enter the number of the choice you want to choose:*

1. *To register for a course*
2. *To deregister for a course*
3. *To list the courses offered*
4. *To list the employees registered for a specific course.*

**JAVA**

**Interface**

**Lab Exercise No:**43

**Exercise Objective(s):***Implementation of an interface*

**Exercise:***Implement Lab Exercise 41using Interfaces.*

**Recommended duration:** *20 Mins*

**Solution Guidance (if applicable):** *NA*

**Lab Exercise No:**44

**Exercise Objective(s):***Implementingmore than one interface.*

**Exercise:***Create two interfaces namely Drawable and Fillable. Create class called Line, Circle, Square and*

*implement following methods through interface.*

*<I>Drawable ------ drawingColor(), thickness()*

*<I>Fillable ---------- fillingColor(), size()*

**Recommended duration:** *20 Mins*

**Solution Guidance (if applicable):** *NA*

**Lab Exercise No:**45

**Exercise Objective(s):***Implementation of an interface*

**Exercise:***In Lab Exercise 39, create a package called finance and within it create an interface called*

*Payable.It should define the getPayInfo() methodthat all the worker classeswillimplement.*

*Now display the details of the monthly pay of the workers.*

**Recommended duration:** *20 Mins*

**Solution Guidance (if applicable):** *NA*

**Lab Exercise No:**46

**Exercise Objective(s):***Implementation of an interface*

**Exercise:***Createa package called bank with the following Interfaces.*

*<Interface> Account*

*<Interface>DepositAcc <Interface>LoanAcc*

*<Interface> Interest*

*<Interface>CreditInterest <Interface>DebitInterest*

1. *<Interface> Account*

*Data members: Four String variables to hold the account type “Savings, Fixed,PersonalLoan,*

*HousingLoan”*

*Methods: createAcc()*

1. *<Interface>DepositAcc*

*Methods: withdraw (), deposit(),getBalance()*

1. *<Interface>LoanAcc*

*Methods: repayPrincipal (),payInterest (),payPartialPrincipal ()*

1. *<Interface>Interest*

*Data members: Four double variables to hold the interest percentage of Savings account, Fixed*

*deposit account,PersonalLoan account and HousingLoan account.*

*Methods: calcInt()*

1. *<Interface>CreditInterest*

*Methods: addMonthlyInt(),addHalfYrlyInt(),addAnnualInt()*

1. *<Interface>DebitInterest*

*Methods: deductMonthlyInt(),deductHalfYrlyInt(),deductAnnualInt()*

*Create a package called BankImpl and create the following classes in it.*

1. *SavingsAcc which implements DepositAcc and CreditInterest*
2. *FDAcc which implements DepositAcc and CreditInterest*
3. *PersonalLoanAcc which implements LoanAcc and DebitInterest*
4. *HousingLoanAcc which implements LoanAcc and DebitInterest*

*Now create a class called MyAccount and create instances of all the accounts and generate appropriate output.*

**Recommended duration:***40Mins*

**Solution Guidance (if applicable):** *NA*

**JAVA**

**Inner class**

**Lab Exercise No:**47

**Exercise Objective(s):***Inner classes*

**Exercise:***Create a class called BankAccount with deposit (), withdraw () and getBalance () methods.*

*Create an inner class called InterestAdder and implement the interest calculations and add*

*the interest to the current balance.*

**Recommended duration:** *20 Mins*

**Solution Guidance (if applicable):** *Interest calculation for a month*

*(Current balance \* 12%)/12*

**Lab Exercise No:**48

**Exercise Objective(s):***Local inner classes*

**Exercise:***Create a class called BankAccount with deposit (), withdraw () and getBalance () methods.*

*Create a local inner class inside the getBalance() method called InterestAdder and implement*

*the interest calculations and add the interest to the current balance.*

**Recommended duration:** *20 Mins*

**Solution Guidance (if applicable):** *Interest calculation for a month*

*(Current balance \* 12%)/12*

**Lab Exercise No:**49

**Exercise Objective(s):***Anonymous inner classes*

**Exercise:***Create a class called BankAccount with deposit (), withdraw () and getBalance () methods.*

*Create an anonymous inner class in the getBalance() method to do the interest calculations*

*and add the interest to the current balance.*

**Recommended duration:** *20 Mins*

**Solution Guidance (if applicable):** *Interest calculation for a month*

*(Current balance \* 12%)/12*

**JAVA**

**Exception Handling**

**Lab Exercise No:**50

**Exercise Objective(s):***syntax*

**Exercise:***In the Lab Exercise 14, change the code such that the numbers are taken as input from the*

*user. Handle the appropriate exceptions.*

**Recommended duration:** *20 Mins*

**Solution Guidance (if applicable):***InputMismatchException, Arithmetic Exception*

**Lab Exercise No:**51

**Exercise Objective(s):***syntax*

**Exercise:***In the Lab Exercise 17, handle the scenarios if the String variable is not initialized.*

**Recommended duration:** *20 Mins*

**Solution Guidance (if applicable):** *Null Pointer Exception*

**Lab Exercise No:**52

**Exercise Objective(s):***syntax*

**Exercise:***Using Lab Exercise 17, catch and demonstratethe required exceptions.*

**Recommended duration:** *20 Mins*

**Solution Guidance (if applicable):***StringIndexOutOfBoundsException*

**Lab Exercise No:**53

**Exercise Objective(s):***syntax*

**Exercise:***Using Lab Exercise 22, catch and demonstrate the required exceptions.*

**Recommended duration:** *20 Mins*

**Solution Guidance (if applicable):***ArrayIndexOutOfBoundsException*

**Lab Exercise No:**54

**Exercise Objective(s):***Exception class methods*

**Exercise:***By using multiple catch blocks, write a class to demonstrate the order of the execution of the*

*catch blocks usingNegativeArraySizeException,ArrayIndexOutOfBoundsException,*

*StringIndexOutOfBoundsException, IndexOutOfBoundsException, NullPointerException,*

*ArithmeticException and print the stack trace for each exception.*

**Recommended duration:** *20 Mins*

**Solution Guidance (if applicable):***NA*

**Lab Exercise No:**55

**Exercise Objective(s):***User-defined exceptions*

**Exercise:***In the Lab Exercise 46, handle the expected exceptions by writing custom defined exceptions.*

**Recommended duration:** *20 Mins*

**Solution Guidance (if applicable):** *NA*

**Lab Exercise No:**56

**Exercise Objective(s):***finally keyword*

**Exercise:***Create a class such that it resets the value of the objects it used to null after its usage in all*

*cases.*

**Recommended duration:** *20 Mins*

**Solution Guidance (if applicable):** *Use finally block*

**Lab Exercise No:**57

**Exercise Objective(s):***finally keyword*

**Exercise:***Create a class such that a method uses the try catch block with the return type of String.*

**Recommended duration:** *20 Mins*

**Solution Guidance (if applicable):** *Use finally block*

**Lab Exercise No:**58

**Exercise Objective(s):***User-defined exceptions*

**Exercise:***Createa class called Employee which asks the user to input the name and the age of a*

*employee. Raise a custom defined exception when the user enters an employee name*

*that has been already entered and raise another exception if the age is negative or less*

*than 18 or greater than 60.*

**Recommended duration:** *20 Mins*

**Solution Guidance (if applicable):***NA*

**JAVA**

**Threads**

**Lab Exercise No:**59

**Exercise Objective(s):***Thread creation and running*

**Exercise:** *Write a program to display the name of the currently running thread.*

**Recommended duration:** *20 Mins*

**Solution Guidance (if applicable):** *NA*

**Lab Exercise No:**60

**Exercise Objective(s):***Thread priority*

**Exercise:***Create a class and spawn 3 threads. Make sure that the Thread2 executes always before*

*Thread3 irrespective of Thread1.*

**Recommended duration:** *20 Mins*

**Solution Guidance (if applicable):** *NA*

**Lab Exercise No:**61

**Exercise Objective(s):***Naming threads, Thread life cycle, and Thread priority*

**Exercise:***Create a class with 4 threads and display a table structured output with the following details*

*every 100 secs 5 times.*

1. *The Id of the thread*
2. *The name of the thread*
3. *Whether the thread is alive or not*
4. *The status of the thread*
5. *The priority level of the thread*

**Recommended duration:** *20 Mins*

**Solution Guidance (if applicable):** *NA*

**Lab Exercise No:**62

**Exercise Objective(s):***Thread creation and running*

**Exercise:** *Create a class and create 2 threads inside*

1. *Non-static inner class*
2. *Local inner class*
3. *Anonymous inner class*

**Recommended duration:** *20 Mins*

**Solution Guidance (if applicable):** *NA*

**Lab Exercise No:**63

**Exercise Objective(s):***Thread life cycle*

**Exercise:***Write a program that picks up a question from an array containing 10 elements and displays*

*the question. It then waits 30 seconds for the user to answer. If user answers it correctly, then*

*it moves to the next question, otherwise it terminates the program.*

**Recommended duration:** *20 Mins*

**Solution Guidance (if applicable):** *NA*

**Lab Exercise No:**64

**Exercise Objective(s):***Thread life cycle*

**Exercise:***Write a class called Clock with two methods Tick and Tock each displaying tick and tock as*

*messages respectively.The outputshould be as follows:*

*Tick1 Tock 1*

*Tick2Tock 2*

*Tick3Tock 3*

*The output goes on till 60.*

**Recommended duration:** *20 Mins*

**Solution Guidance (if applicable):** *NA*

**Lab Exercise No:**65

**Exercise Objective(s):***Synchronization*

**Exercise:***Write an application to simulate the vehicles crossing a bridge and a toll plaza on a highway.*

*For the purpose of this exercise, simulate the environment for five vehicles that are*

*approaching the bridge and the toll booth. The vehicles are numbered from one to five. The*

*vehicles should approach the bridge and the toll booth in sequential order. The toll booth can*

*only deal with one vehicle at a time. This application should print a message every time when*

*a vehicle crosses the bridge and another message when a vehicle crosses the toll booth along*

*with the vehicle number.*

**Recommended duration:** *20 Mins*

**Solution Guidance (if applicable):***The simulation is performed by having one thread for each vehicle.*

*and consists of four classes:*

*-Vehicle: Simulate the behavior of a vehicle*

*-Bridge:Simulate the behavior of a Bridge*

*-TollBooth: Simulate the behavior of a TollBooth*

*-Simulate: creates the vehicles and controls the simulation.*

**Lab Exercise No:** 66

**Exercise Objective(s):***Thread priority*

**Exercise:***Consider the following scenario. Whenever a hen lays an egg its owner sells the egg to a shop.*

*In the last 4 months the owner has gained Rs. 100 by selling eggs in the rate of Rs.2 per egg.*

*Display the following messages*

*Hen Laid the Egg – 1*

*Owner gained Rs 2*

*Hen Laid the Egg – 2*

*Owner gained Rs 4*

*…*

*…*

*…*

*…. So on.*

**Recommended duration:**  *20 Mins*

**Solution Guidance (if applicable):** *This should be done based on the Producer-Consumer logic.*

**JAVA**

**Immutable and mutable strings and primitives objects**

**Lab Exercise No:**67

**Exercise Objective(s):***Strings, Immutability*

**Exercise:***Write a program that gets a name (name appended with surname) as a input and*

*Computestheinitials(First letter of each word) from the full name and displays them.*

**Recommended duration:** *20 Mins*

**Solution Guidance (if applicable):** *For Eg: If the name is James Gosling,*

*Then the Output should be: JG*

**Lab Exercise No:**68

**Exercise Objective(s):***Strings, Immutability*

**Exercise:***Write a program to check whether the given strings are an anagram or not. An anagram is a*

*word or a phrase made by transposing the letters of another word or phrase;for example,*

*"Parliament" is an anagram of "partial men". The program should ignore white*

*space and punctuation.*

**Recommended duration:** *20 Mins*

**Solution Guidance (if applicable):** *NA*

**Lab Exercise No:**69

**Exercise Objective(s):***Strings, Immutability*

**Exercise:***Create an array of 10 students’ names. Convert the array as a single object and print the*

*names.*

**Recommended duration:** *20 Mins*

**Solution Guidance (if applicable):** *NA*

**JAVA**

**java.util classes**

**Lab Exercise No:**70

**Exercise Objective(s):***Working with Dates*

**Exercise:***Display the current time with AM-PM format. Fine no change*

**Recommended duration:** *20 Mins*

**Solution Guidance (if applicable):** *NA*

**Lab Exercise No:**71

**Exercise Objective(s):***Working with Dates*

**Exercise:***Display your date of birth in a format such that the month is printed in alphabets.*

**Recommended duration:** *20 Mins*

**Solution Guidance (if applicable):** *NA format soln*

**Lab Exercise No:** 72

**Exercise Objective(s):***Working with Dates*

**Exercise:***Consider a movie is released today. Display the silver jubilee, golden jubilee, diamond jubilee*

*dates of the movie assuming that it will run successfully.*

**Recommended duration:**  *20 Mins*

**Solution Guidance (if applicable):** *Silver Jubilee 25*

*Golden Jubilee 50*

*Diamond Jubliee 60*

*Platinum Jubliee 75*

**Lab Exercise No:**73

**Exercise Objective(s):***Working with Dates*

**Exercise:***Display your day of birth by using the Calendar class.*

**Recommended duration:** *20 Mins*

**Solution Guidance (if applicable):** *NA*

**Lab Exercise No:**74

**Exercise Objective(s):***Working with Dates*

**Exercise:***Displaythe current date and time in three different formats of different countries.*

**Recommended duration:** *20 Mins*

**Solution Guidance (if applicable):** *NA soln format*

**Lab Exercise No:**75

**Exercise Objective(s):***Math class*

**Exercise:***Display the cube roots of the list of prime numbers till 100.*

**Recommended duration:** *20 Mins*

**Solution Guidance (if applicable):** *NA*

**Lab Exercise No:**76

**Exercise Objective(s):***Math class*

**Exercise:** *Write a program that accepts a double value as input and displays the floor value and ceil*

*value of the same.*

**Recommended duration:** *20 Mins*

**Solution Guidance (if applicable):** *NA*

**Lab Exercise No:**77

**Exercise Objective(s):***StringTokenizer*

**Exercise:***In the Lab Exercise* **17**, print that string as separate tokens.

**Recommended duration:** *20 Mins*

**Solution Guidance (if applicable):** *NA*

**JAVA**

**Collections and Generics**

**Lab Exercise No:**78

**Exercise Objective(s):***List interface*

**Exercise:***Create a collection that will contain the names of the days in a week. Print the collection.*

*Display the length of the collection and convert the collection into an array and print it.*

**Recommended duration:** *20 Mins*

**Solution Guidance (if applicable):** *NA*

**Lab Exercise No:**79

**Exercise Objective(s):***Classes implementing Set interface*

**Exercise:***Write a program that will accept 10 numbers from the user as input and store it in a collection.*

*The numbers should not be duplicated. Also get the index position along with the number*

*where it has to be stored.*

**Recommended duration:** *20 Mins*

**Solution Guidance (if applicable):** *NA*

**Lab Exercise No:**80

**Exercise Objective(s):***Classes implementing Map interface*

**Exercise:***Write a program to implement a telephone directory. Display the details.*

**Recommended duration:** *20 Mins*

**Solution Guidance (if applicable):**  *Name Phone no*

*ABC 1234*

*DEF 5678*

**Lab Exercise No:**81

**Exercise Objective(s):***Classes implementing Map interface*

**Exercise:***Create a program to depict the usage of the dictionary where words along with the meanings*

*are stored. When the user gives a word, its meaning should be displayed.*

**Recommended duration:** *20 Mins*

**Solution Guidance (if applicable):** *NA*

**Lab Exercise No:** 82

**Exercise Objective(s):***List interface*

**Exercise:***Create a program to demonstrate the stack methods push () and pop ()by using a real time*

*example.*

**Recommended duration:**  *20 Mins*

**Solution Guidance (if applicable):**  *NA*

**Lab Exercise No:**83

**Exercise Objective(s):***Collections and arrays classes*

**Exercise:***Create a class called CD whose attributes are Title and singer. Arrange the CDs in ascending*

*order based on the singer name.*

**Recommended duration:** *20 Mins*

**Solution Guidance (if applicable):** *NA*

**Lab Exercise No:**84

**Exercise Objective(s):***Understanding generics*

**Exercise:***Design a class that acts as a library for the following kinds of media: book, video, and*

*newspaper. Provide one version of the class that uses generics and one that does not.*

*The class should be with add and retrieve methods in order to add and retrieve values*

*from the library.*

**Recommended duration:** *20 Mins*

**Solution Guidance (if applicable):** *NA*

**Lab Exercise No:**85

**Exercise Objective(s):***Collections and arrays classes*

**Exercise:***Create a class called BookStore with fields Bookid and Bookname. Sort using comparator and*

*display the output by sorting based on book name.*

**Recommended duration:** *20 Mins*

**Solution Guidance (if applicable):** *NA*

**JAVA**

**IO**

**Lab Exercise No:**86

**Exercise Objective(s):***java.io.File*

**Exercise:** *Write a program that creates a new file called batch mates and store your entire batch mates name in it and display the details.*

**Recommended duration:***20 Mins*

**Solution Guidance (if applicable):***NA*

**Lab Exercise No:**87

**Exercise Objective(s):***java.io.File*

**Exercise:***Write aprogram that counts the number of times a particular character(character should be got as a input from the user), appears in a file. Use the file that is created in Lab Exercise 86.*

**Recommended duration:** *20 Mins*

**Solution Guidance (if applicable):** *NA*

**Lab Exercise No:**88

**Exercise Objective(s):***java.io.File*

**Exercise:***Write a program that creates a new file called my batch in a directory named Batch and copy the contents of the batch mates file in it.*

**Recommended duration:** *20 Mins*

**Solution Guidance (if applicable):** *NA*

**Lab Exercise No:**89

**Exercise Objective(s):***java.io.File*

**Exercise:***Create a program to list files or subdirectories in a directory.*

**Recommended duration:** *20 Mins*

**Solution Guidance (if applicable):** *NA*

**Lab Exercise No:**90

**Exercise Objective(s):***Character streams classes*

**Exercise:***Create a program that accepts the 5 student roll no, name and grade through Console class. Create a properties file called class.properties and store the data by giving appropriate comments.*

**Recommended duration:** *20 Mins*

**Solution Guidance (if applicable):** *For Eg, the properties file should look like,*

*#Student 1*

*Student1:31,Charles,A*

**Lab Exercise No:**91

**Exercise Objective(s):***Character streams classes*

**Exercise:***Create a CSV file with 5 movie names along with hero and heroine names. Write a program to read from the CSV file and display it to the user.*

**Recommended duration:** *20 Mins*

**Solution Guidance (if applicable):** *NA*

**Lab Exercise No:**92

**Exercise Objective(s):***java.io.File*

**Exercise:***Write a program that accepts the name of your batch mate as an input and displays the line number of that name in the file created in Lab Exercise 86. If any other name is been inputted display appropriate message.*

**Recommended duration:** *20 Mins*

**Solution Guidance (if applicable):** *NA*

**Lab Exercise No:**93

**Exercise Objective(s):***java.io.File*

**Exercise:***Write a program that displays the total* number of lines, number of words and number of characters in the file *created in Lab Exercise 86.*

**Recommended duration:** *20 Mins*

**Solution Guidance (if applicable):** *NA*

**Lab Exercise No:**94

**Exercise Objective(s):***Serialization*

**Exercise:***Write a program to create a class called Car with Model, Year of making details. Store and retrieve this information using Binary Serialization.*

**Recommended duration:** *20 Mins*

**Solution Guidance (if applicable):** *NA*

**Lab Exercise No:**95

**Exercise Objective(s):***Serialization*

**Exercise:***Create an object called employee whose attributes are emp\_id, emp\_name and emp\_sal. Write a program to Serialize and deserializethe employee object except for the emp\_sal attribute.*

**Recommended duration:** *20 Mins*

**Solution Guidance (if applicable):** *Use transient*

**JDBC**

**Lab Exercise No:** 96

**Exercise Objective(s):** *Working with native drivers to insert, update and delete and search*

**Exercise:** *Write a program to create a table called training with the following attributes,*

1. *Sap\_ID*
2. *Stud\_name*
3. *Stream*
4. *Percentage*

*The program should insert all your batch mates’ details and display the same.*

**Recommended duration:** *20 Mins*

**Solution Guidance (if applicable):** *NA*

**Lab Exercise No:** 97

**Exercise Objective(s):** *Statement and Result Set*

**Exercise:** *Write a program to Read, Update and Delete any record from the Element table. The table Element has following fields (Atomic weight, Name and Chemical symbol). The input should be provided through Command Line Argument along with the appropriate data.*

**Recommended duration:** *20 Mins*

**Solution Guidance (if applicable):** *NA*

**Lab Exercise No:** 98

**Exercise Objective(s):** *Advanced Result Set*

**Exercise:** *Display the set of elements from the Element table in Lab Exercise no 120 and the make the records only readable.*

**Recommended duration:** *20 Mins*

**Solution Guidance (if applicable):** *NA*

**Lab Exercise No:** 99

**Exercise Objective(s):** *Working with Prepared Statement*

**Exercise:** *From the Lab exercise 16, create a Student table and insert the values using prepared statement and display the table.*

**Recommended duration:** *20 Mins*

**Solution Guidance (if applicable):** *NA*

**Lab Exercise No:** 100

**Exercise Objective(s):** *Working with Prepared Statement*

**Exercise:** *From the Lab Exercise 122, add two more student details to the table and display only the RollNo, StudName and MarksInMaths.*

**Recommended duration:** *20 Mins*

**Solution Guidance (if applicable):** *NA*

**Lab Exercise No:** 101

**Exercise Objective(s):** *Working with PreparedStatement*

**Exercise:** *List the trainees from the training table (Lab Exercise no 119). Display the first, nth and last record in that list.*

**Recommended duration:** *20 Mins*

**Solution Guidance (if applicable):** *NA*

**Lab Exercise No:** 102

**Exercise Objective(s):***Invoking stored procedure-Callable Statement*

**Exercise:** *From Lab Exercise 25, create a table called Employee. Add a column called salary. Create a procedure called updateprocedure which will update the salary of an employee. Update the salary of an employee by calling the procedure and passing the employee id of a particular employee.*

**Recommended duration:** *20 Mins*

**Solution Guidance (if applicable):** *NA*

**Lab Exercise No:** 103

**Exercise Objective(s):***Transaction with JDBC*

**Exercise:** *Create a table called Account with AccNo, AccName, AccType and Balance. Now consider a transfer of money from AccNo 1 to AccNo2. Display the AccNo and Balance before the transfer and after the transfer. Try both the scenarios of transfer is successful and transfer is failure.*

**Recommended duration:** *20 Mins*

**Solution Guidance (if applicable):** *NA*

**Lab Exercise No:** 104

**Exercise Objective(s):***JdbcRowSet*

**Exercise:** *Using Lab Exercise 119, delete 2 trainees name from the table using JdbcRowSet.*

**Recommended duration:** *20 Mins*

**Solution Guidance (if applicable):** *NA*

**JDBC**

**Lab Exercise No:** 105

**Exercise Objective(s):** *Working with native drivers to insert, update and delete and search*

**Exercise:** *Write a program to create a table called training with the following attributes,*

1. *Sap\_ID*
2. *Stud\_name*
3. *Stream*
4. *Percentage*

*The program should insert all your batch mates’ details and display the same.*

**Recommended duration:** *20 Mins*

**Solution Guidance (if applicable):** *NA*

**Lab Exercise No:** 106

**Exercise Objective(s):** *Statement and Result Set*

**Exercise:** *Write a program to Read, Update and Delete any record from the Element table. The table Element has following fields (Atomic weight, Name and Chemical symbol). The input should be provided through Command Line Argument along with the appropriate data.*

**Recommended duration:** *20 Mins*

**Solution Guidance (if applicable):** *NA*

**Lab Exercise No:** 107

**Exercise Objective(s):** *Advanced Result Set*

**Exercise:** *Display the set of elements from the Element table in Lab Exercise no 120 and the make the records only readable.*

**Recommended duration:** *20 Mins*

**Solution Guidance (if applicable):** *NA*

**Lab Exercise No:** 108

**Exercise Objective(s):** *Working with Prepared Statement*

**Exercise:** *From the Lab exercise 16, create a Student table and insert the values using prepared statement and display the table.*

**Recommended duration:** *20 Mins*

**Solution Guidance (if applicable):** *NA*

**Lab Exercise No:** 109

**Exercise Objective(s):** *Working with Prepared Statement*

**Exercise:** *From the Lab Exercise 122, add two more student details to the table and display only the RollNo, StudName and MarksInMaths.*

**Recommended duration:** *20 Mins*

**Solution Guidance (if applicable):** *NA*

**Lab Exercise No:** 110

**Exercise Objective(s):** *Working with PreparedStatement*

**Exercise:** *List the trainees from the training table (Lab Exercise no 119). Display the first, nth and last record in that list.*

**Recommended duration:** *20 Mins*

**Solution Guidance (if applicable):** *NA*

**Lab Exercise No:** 111

**Exercise Objective(s):***Invoking stored procedure-Callable Statement*

**Exercise:** *From Lab Exercise 25, create a table called Employee. Add a column called salary. Create a procedure called updateprocedure which will update the salary of an employee. Update the salary of an employee by calling the procedure and passing the employee id of a particular employee.*

**Recommended duration:** *20 Mins*

**Solution Guidance (if applicable):** *NA*

**Lab Exercise No:** 112

**Exercise Objective(s):***Transaction with JDBC*

**Exercise:** *Create a table called Account with AccNo, AccName, AccType and Balance. Now consider a transfer of money from AccNo 1 to AccNo2. Display the AccNo and Balance before the transfer and after the transfer. Try both the scenarios of transfer is successful and transfer is failure.*

**Recommended duration:** *20 Mins*

**Solution Guidance (if applicable):** *NA*

**Lab Exercise No:** 113

**Exercise Objective(s):***JdbcRowSet*

**Exercise:** *Using Lab Exercise 119, delete 2 trainees name from the table using JdbcRowSet.*

**Recommended duration:** *20 Mins*

**Solution Guidance (if applicable):** *NA*