JAVA

**Introduction to Java**

**Lab Exercise No:** 1

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

**Exercise:** *Write a java program to display the following*

*\* \* \**

*\* \**

*\**

1. *Use print and println statements.*
2. *The class file of this program should be automatically placed inside “Design” folder while compiling.*
3. *Display the version used for compiling.*

**Recommended duration:** *15 Mins*

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

**Lab Exercise No:** 2

**Exercise Objective(s):** *Compilation and execution from command line, Classpath*

**Exercise:**

*public class A{*

*public static void main(String[] args)*

*{*

*String curDir = System.getProperty("user.dir");*

*System.out.println(curDir);*

*}*

*}*

*This code is in C:\MyJava folder. If this code is executed from C:\MyJava, it prints the current directory.*

*What command will make sure that this code can be executed from any location?*

**Recommended duration:** *10 Mins*

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

[**Lab Exercise No:** 3](http://java//Ex3.txt.docx)

**Exercise Objective(s):** *Java doc*

**Exercise:** *Provide java doc comments for the class created in the previous exercise and generate HTML document files. Use documentation annotation.*

**Recommended duration:** *10 Mins*

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

**Basic elements of Java**

**Lab Exercise No:**  4

**Exercise Objective(s):** *Primitive data types and their range, Variables, Simple operators*

**Exercise:** *Write a program to do the following,*

1. *Get two numbers as input from the user through console and swap the values of two numbers without using a temporary variable and display the same.*
2. *Determine whether the given year is leap year or not (Read the input through console).*

**Recommended duration:** *30 Mins*

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

**Lab Exercise No:** 5

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

**Exercise:**

*Tax slabs for general*

*0 to 180000 No tax*

*180001 to 500000 10%*

*500001 to 800000 20%*

*Above 800000 30%*

*Income tax slabs 2011-2012 for Women*

*0 to 190000 No tax*

*190001 to 500000 10%*

*500001 to 800000 20%*

*Above 800000 30%*

*Write if statements to achieve this.*

*Make sure that you indent the code well so that it is readable.*

**Recommended duration:** *20 Mins*

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

**Lab Exercise No:** 6

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

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

| ***Product No*** | ***Product Code*** | ***Retail Price*** |
| --- | --- | --- |
| *1* | *A* | *22.50* |
| *2* | *B* | *44.50* |
| *3* | *C* | *9.98* |

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

*a) Product number or code (Code is not case sensitive)*

*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:** *20 Mins*

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

**Lab Exercise No:** 7

**Exercise Objective(s):** *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 console. The program should display how many gross, how many dozen, and how many left over eggs the user has.*

*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*

**Recommended duration:** *20 Mins*

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

**Lab Exercise No:** 8

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

**Exercise:** *Write a program to display whether the given number is palindrome or not. Also, check whether the number is a prime number. If it is a prime number, display the number along with the alphabet ‘p’ appended to it.*

**Recommended duration:** *20 Mins*

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

**JAVA**

**Classes and Methods Part-1**

**Lab Exercise No:** 9

**Exercise Objective(s):** *Construction of an object, instance and class member variables*

**Exercise:** *Create a class called Marks with the following details: regNo, MarksInEng, MarksInMaths and MarksInScience. Write getters and setters for the all variables. Use the class Student which is already created for getting the details of the student. Create a class called Standard with 5 students’ details and write separate method for each of the following tasks and invoke the same.*

*a) To display the entire regNo and the name of the students in the class in the ascending order of regNo.*

*b) To display the regNo and the name of the student who has got the highest percentage.*

*c) To display the regNo and the name of the student who scored highest mark in mathematics.*

*d) To display the regNo and the name of the student in the ascending order of the total marks in mathematics and science alone.*

*e) To display the regNo, name, total marks, percentage and rank of all the students in the descending order of rank.*

**Recommended duration:** *30 Mins*

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

**Lab Exercise No:** 10

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

**Exercise:** *Create a Stack class that can hold any number of integers. Provide methods like push and pop. Use appropriate OOPs principles.*

**Recommended duration:** *30 Mins*

**Solution Guidance (if applicable):** *Encapsulate an int array with some predefined size. During push operation, if there is an overflow, then the array size should be increased. Make sure that values are not lost in process of doing this.*

**Lab Exercise No:** 11

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

**Exercise:** *Create an ArrayList that can hold integers. Use methods to insert, delete and find values in the ArrayList.*

**Recommended duration:** *30 Mins*

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

**Lab Exercise No:** 12

**Exercise Objective(s):** *Multi-dimensional arrays*

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

**Recommended duration:** *30 Mins*

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

**Lab Exercise No:** 13

**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 length of the String.*
8. *Check whether the String matches to “The quick brown Fox jumps over the lazy Dog”.*
9. *Replace the word “The” with the word “A”.*
10. *Split the above string into two such that two animal names do not come together.*
11. *Print the animal names alone separately from the above string.*
12. *Print the above string in completely lower case.*
13. *Print the above string in completely upper case.*
14. *Find the index position of the character “a”.*
15. *Find the last index position of the character “e”.*

**Recommended duration:** *30 Mins*

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

**Lab Exercise No:** 14

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

**Exercise:** *Given an array of 10 students’ names. Convert the array as a single string and print it.*

**Recommended duration:** *15 Mins*

**Solution Guidance (if applicable):** *Hint: Efficient use of memory is the focus here*

**Lab Exercise No:** 15

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

**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, “Ate" is an anagram of “Eat". The program should ignore white space and punctuation.*

**Recommended duration:** *30 Mins*

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

**JAVA**

**Classes and Methods Part-2**

**Lab Exercise No:** 16

**Exercise Objective(s):** *Passing values to methods*

**Exercise:** *You have created a Student class in the previous session. Create a student array and populate it with student objects. Have a static method called sort. Pass array of students to the sort method. Sort the array in the method say based on id. Do you find the array in the sorted form when the sort () method call returns back to the main method?*

**Recommended duration:** *30 Mins*

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

**Lab Exercise No:** 17

**Exercise Objective(s):** *Formatting*

**Exercise:** *Write a program that accepts the user’s full name , gets the Unicode char for each letter and passes these numbers to the method called generatePassword(). The method generatePassword() should reverse each number and then combine all the reversed numbers into one. If the number is greater than 5 digits then number must be divided by 5 until 5 digits is reached. Finally this number is displayed it in its octal and hexadecimal notation.*

**Recommended duration:** *60Mins*

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

**Lab Exercise No:** 18

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

**Exercise:** *Create a class called Calculator which has 4 different methods add, diff, mul and div that accept 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 an object to access these methods and invoke these methods with different type of numbers and display the result in the corresponding methods.*

**Recommended duration:** *30Mins*

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

**Lab Exercise No:** 19

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

**Exercise:** *Write a class called AppUser that prompts the user for the following information*

1. *Database URL*
2. *Property file name*
3. *User name*
4. *Password*

*Since these are Database URL and Property file name are common for all users and hence they have to be initialized only once when the code is executed.*

*User name and password are specific to each AppUser object.*

*Test the application by creating 2 users and print all the details entered by the user.*

**Recommended duration:** *15 Mins*

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

**JAVA**

**Packages**

**Lab Exercise No:** 20

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

**Exercise:**

* *Create a new project in which create a package named org.animals. In that create various classes like Lion, Monkey, Elephant. In each class create data members like color, weight and age. Create methods like isVegetarian, canClimb, getSound.*
* *Create another project and in that create a package called zoo and create a class called VandalurZoo 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.*

**JAVA**

**Inheritance**

**Lab Exercise No:** 21

**Exercise Objective(s):** *Simply inheriting from a class*

**Exercise:**

* *Create a class called Vehicle. Create subclasses Truck, Bus and Car. Add common methods in the base class and specific methods in the corresponding subclasses. Create a class called Road and create objects for Truck, Bus and Car and display the appropriate messages. Also, in the Vehicle class constructor, initialize the variables color, no of wheels and model. Give appropriate values for these variables from the invoking subclass.*

**Recommended duration:** *20Mins*

**Solution Guidance (if applicable):** *None*

**Lab Exercise No:** 22

**Exercise Objective(s):** *working with classes inheriting from a class and using super class members.*

**Exercise:**

* *Create a class called Inventory in a package stock. This class has data members called quantity and lowOrderLevelQuantity. Two classes that inherit from this class -Accessories and Laptops, are in package called material. The lowOrderLevelQuantity for laptops is 3, while lowOrderLevelQuantity for Accessories is 5. Apart from these members, Accessories and Laptops also have members describing them and a unique id.*
* *Create 5 laptops and 10 Accessories objects. The quantity member must add up accordingly in the individual classes.*
* *Create an Order class and have customers place orders. If the ordered quantity is available then Invoice should be generated. If the quantity is below lowOrderLevelQuantity then a RequestForMaterial (RFM) must be generated.*

**Recommended duration:** *45 Mins*

**Solution Guidance (if applicable):** *None*

**Lab Exercise No:** 23

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

**Exercise:**

* *Create a class called Worker. Write classes DailyWorker and SalariedWorker that inherit from Worker. Every worker has a name and a salary. Write method pay() 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 worked hours are. Create a few different types of workers and print all the details of the workers(name, salary and D/W (indicating the type of worker)) in sorted order of the salary.*

**Recommended duration:** *45 Mins*

**Solution Guidance (if applicable):** *None*

**Lab Exercise No:** 24

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

**Exercise:**

* *The Charity Collection Box contains money in different currencies - dollars-cents or pounds-pence or rupees-paise. All of these currencies have notes and coins. The note and coin numbers are counted when they are added based on their value (that is number of 5 rupee notes, or $1 dollar note).*
* *A super class representing Currency is created with different denomination for of notes and coins. Subclass Dollar, Pound and Rupee has conversion methods to rupees, print() and compute().*

*Create class called CollectionBox that allows entry of these currencies in terms of number of notes and coins of different denomination. Create a display method that allows any of these currency types and displays the total amount collected in terms of Rupees. (Assume1 dollar= Rs. 50 and 1 pound = 1.6232 U.S. dollars.*

**Recommended duration:** *45 Mins*

**Solution Guidance (if applicable):** *None*

**Lab Exercise No:** 25

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

**Exercise:**

* *An application interacts with databases of different types. Every database object has user name, password and a url.*
* *Databases classes- Oracle, SQLServer and MySql are created.*
* *Only10 database objects are allowed at a time. And these are stored in an array that can represent any of the above objects. No further subclasses can be created to prevent the numbers going beyond 10.*
* *Provide a display method that will print the details of the database objects in the array.*
* *Test the application by creating 10 objects and display it and then creating 11th object.*

**Recommended duration:** *30 Mins*

**Solution Guidance (if applicable):**  *GetInstance() method that returns instances of the object is overridden in the individual classes. This will make sure that number of instances does not go beyond 10.*

**Lab Exercise No:** 26

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

**Exercise:**

*In the previous exercise, determine which class and method can be made abstract class or method. Test the application*

**Recommended duration:** *15 Mins*

**Solution Guidance (if applicable):**  *None*

**Lab Exercise No:** 27

**Exercise Objective(s):** *Overriding Object class methods*

**Exercise:**

* *In the worker exercise, instead of printing individual attributes like name, salary and so on, if the object is printed automatically the details must be printed. Also two workers are same if their names are same. Therefore before printing salary report, a check needs to made to see if duplicate workers have been entered. If so, the duplicates must be removed from the list.*

**Recommended duration:** *30 Mins*

**Solution Guidance (if applicable):**  *None*

**Lab Exercise No:** 28

**Exercise Objective(s):** *Overriding Object class methods*

**Exercise:**

* *Overriding hashcode() and equals() method for Student such that all the ids that are prime and even are goes in one bucket, all the ids that are prime and odd are in another and rest in yet another.*
* *Make sure that hashcode() and equals() method follow the contract specified in the documentation.*

**Recommended duration:** *30 Mins*

**Solution Guidance (if applicable):**  *None*

**Lab Exercise No:** 29

**Exercise Objective(s):** *Overriding Object class methods*

**Exercise:**

* *A class Connection maintains attributes database url, user name and password. Class needs to maintain the number of count of Connection class. Every time a connection object is created the count must be incremented and every time it is set to null count must be decremented and object must be garbage collected explicitly by the code. At any point, there must be only 10 connection object in the memory. Write a java class to achieve this.*

**Recommended duration:** *30 Mins*

**Solution Guidance (if applicable):**  *None*

**JAVA**

**Interface**

**Lab Exercise No:** 30

**Exercise Objective(s):** *using interface constants*

**Exercise:**

* *A scientific application needs to use several constants through out the application. It was decided to create an interface called PhysicalConstants. The constants that needs to be set are* 
  + *Speed of light in vacuum (C):299 792 458 m/s*
  + *Gravitational constant (G): 6.674 28×10−11 m3 kg−1 s−2*
  + *Standard Gravitational Acceleration(g) : 9.806 65 m/s2*
* *Use these constants for a class that has following functions.* 
  + *E= MC2*
  + *F=G (m1 m2)/r2*
  + *d=0.5 gt*

*Make sure that constants names are used without repeating interface names with constants.*

**Recommended duration:** *30 Mins*

**Solution Guidance (if applicable):** *use static imports*

**Lab Exercise No:** 31

**Exercise Objective(s):** *using Comparable*

**Exercise:**

*Given a unsorted list of 10 athletes nominated for the coaching class, provide a way for the coach to search for the athlete name and provide grades. Finally print the list of athletes’ names with their grade in the sorted order of their names.*

**Recommended duration:** *30 Mins*

**Solution Guidance (if applicable):** *Use Arrays class methods*

**Lab Exercise No:** 32

**Exercise Objective(s):** *using Comparator*

**Exercise:**

* *Provide a way to print list of athletes sorted by the grade as well.*

**Recommended duration:** *30 Mins*

**Solution Guidance (if applicable):** *None*

**Lab Exercise No:** 33

**Exercise Objective(s):** *using abstract classes and interfaces*

**Exercise:**

* *Shape abstract class has 2 abstract methods*
  + *area()*
  + *volume()*
* *Classes Cube, Rectangle, Triangle and Sphere are created . For Rectangle, Triangle volume returns -1.*
* *Shapes that implement volume must be of type Spatial which is a marker interface.*
* *The user enters 5 shape objects which is stored in an array.*
* *Finally, all the Shape objects are printed. Only for Shape object of Spatial type, volume is printed.*

*Hint:*

*Triangle Area = 1/2 of the base X the height,*

*Rectangle Area: l X w*

*Sphere Area= 4pr2  , Volume = 4/3 pr3*

*Cube Area = 2lw + 2lh + 2wh Volume = l X w X h*

*Where l is length, w is width and h is height*

**Recommended duration:** *30 Mins*

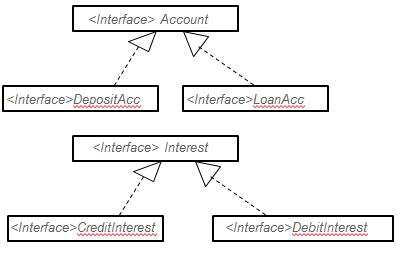
**Solution Guidance (if applicable):** *None*

**Lab Exercise No:** 34

**Exercise Objective(s):** *working with multiple interfaces*

**Exercise:**

*Create a package called bank with the following Interfaces.*

**

*Interface> Account*

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

*PersonalLoan,HousingLoan”*

*Methods: createAcc()*

*<Interface>DepositAcc*

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

*<Interface>LoanAcc*

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

*<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()*

*<Interface>CreditInterest*

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

*<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:** *1 hrs*

**Solution Guidance (if applicable):** *None*

**JAVA**

**Nested classes**

**Lab Exercise No:** 35

**Exercise Objective(s):** *using non-static inner class*

**Exercise:**

* *Create an AddressBook class encapsulating name (String), tempAdd (Address), premAdd (Address) and phone number(long). Provide appropriate getter and setter methods.*
* *Address class encapsulates house address– name, street address, city, state.*
* *The Address class must be available only to AddressBook class.*
* *Make sure that name assigned to AddressBook object is same as name assigned to Address object.*
* *Test the application by creating AddressBook object and printing address and email id assigned to AddressBook object.*

**Recommended duration:** *1 hour*

**Solution Guidance (if applicable):** *None*

**Lab Exercise No:** 36

**Exercise Objective(s):** *using non-static inner class*

**Exercise:**

*Part 1: Create an abstract class called BalanceComputer that has one implemented static method called getBalanceComputer (char type) and one abstract method getBalance(). The getBalanceComputer() method returns BalanceComputer object with one of the formula (listed below) implemented for getBalance() based on the type of account: current (C)or savings(S).*

*For current acct formula is :*

*amt (end of the year)= principal amount+ (principal amount \* rate \* time )/100*

*For Savings formula is :*

*amt (calculated quarterly)= principal amount (1+ rate/4)4t*

*Hint: use anonymous inner class.*

*Part 2: Create another class called BankAccount with acct number, acct type, month of deposit, deposit (), withdraw (), and getBalance () methods. The getBalance() must determine the acct type and get the appropriate BalanceComputer object . Using this object and the balances (at the end or quarter or year based on the account type) from list of transactions (deposit/ withdraw), getBalance() must compute the amount and return the same. Test this application.*

**Recommended duration:** *1 hour*

**Solution Guidance (if applicable):** *None*

**JAVA**

**Exception Handling**

**Lab Exercise No:** 37

**Exercise Objective(s)**: Simple exception handling

**Exercise:**

* *In the Calculator program(Slide no 36 Classes and Methods part 2), modify the program such that the numbers are taken as input from the user. Handle the appropriate exceptions.*

**Recommended duration:** *15 mins*

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

**Lab Exercise No:** 38

**Exercise Objective(s):** Simple exception handling

**Exercise:**

* *A comma separated list containing pairs of topic name, time in hours ( Java 14, JEE 10, JME 12) will be entered in the command line arguments. If a day consists of 8 hours, list out the topics that will be covered day-wise. Catch all the possible exceptions.*

**Recommended duration:** *30 mins*

**Solution Guidance (if applicable):**

**Lab Exercise No:** 39

**Exercise Objective(s):** *delegating exception handling*

**Exercise:**

* *In the previous exercise in slide 18, modify the program such that the exceptions are not handled in the add, diff, mul and div methods. The exception handling should be delegated to the caller method.*

**Recommended duration:** *15 mins*

**Solution Guidance (if applicable):**

**Lab Exercise No:** 40

**Exercise Objective(s):** *using clone()*

**Exercise:**

*Create a class called LinkedList. Override and provide public access to clone method for this class.*

*Hint: LinkedList class encapsulates Node object that has attributes :num (int) and next (Node).*

*Since LinkedList has reference, make sure that clone is not Object’s clone() that does bitwise copy only.*

**Recommended duration:** *40 mins*

**Solution Guidance (if applicable):**

**Lab Exercise No:** 41

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

**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:** *30 mins*

**Solution Guidance (if applicable):**

**Lab Exercise No:** 42

**Exercise Objective(s):** *using exception handling and exception wrapping*

**Exercise:**

* *Create a class called Employee that asks the user to input the name and the age of an employee. Raise a custom defined exception when the user enters an employee name that has already been entered and raise another exception if the age is negative or less than 18 or greater than 60. If there is any occurrence of InputMismatchException and NumberFormatException, throw those also as user defined exceptions.*

**Recommended duration:** *30 mins*

**Solution Guidance (if applicable):**  *use exception wrapping*

**JAVA**

**Threads**

**Lab Exercise No:** 43

**Exercise Objective(s):** *using non-static inner class*

**Exercise:**

* *The main method waits to get input from the users until it is terminated. The input that it receives is any string. As soon as the main method receives the input it delegates the assignment of creating a password to a thread. The thread generates a random number and appends this number to the string that is passed and displays the password. Write a java program to do this.*

**Recommended duration:** *45 mins*

**Solution Guidance (if applicable):**  *Use java.util.Random class to generate random numbers.*

**Lab Exercise No:** 44

**Exercise Objective(s):** *using non-static inner class*

**Exercise:**

* *Implement the previous exercise using Runnable.*

**Recommended duration:** *15 mins*

**Solution Guidance (if applicable):**

**Lab Exercise No:** 45

**Exercise Objective(s):** *using non-static inner class*

**Exercise:**

* *A prompt asking a question appears for which user is given 1 minute. If user answers the question before 1 minute then “Congratulations!” is displayed. Otherwise “Better Luck Next Time” is displayed.*
* *Write a program to implement the above scenario.*

**Recommended duration:** *30 mins*

**Solution Guidance (if applicable):**

**Lab Exercise No:** 46

**Exercise Objective(s):** *using non-static inner class*

**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:** *45 mins*

**Solution Guidance (if applicable):**

**Lab Exercise No:** 47

**Exercise Objective(s):** *using non-static inner class*

**Exercise:**

* *Write a bank class that has an array of account objects. The method transfer allows transfer of money from one account to another account. Using the hints to avoid deadlocks from the previous slide implement the transfer method.*
* *Test the application by creating two threads that simultaneously transfers money form accounts 11111111 to account 22222222 and vice versa.*

**Recommended duration:** *30 mins*

**Solution Guidance (if applicable):**

**Lab Exercise No:** 48

**Exercise Objective(s):** *using non-static inner class*

**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:** *45 mins*

**Solution Guidance (if applicable):**

**JAVA**

**Immutable and mutable strings and primitive objects**

**Lab Exercise No:** 49

**Exercise Objective(s):** *using non-static inner class*

**Exercise:**

* *Consider the following rules for the evaluation of a physical fitness for a*

*post in a company where only men can apply.*

| ***Age*** | ***Weight in pounds*** | ***Height in inches*** |
| --- | --- | --- |
| *Below 20* | *Not Eligible* | *Not Eligible* |
| *20 to 30* | *155 to 175* | *5’2” to 5’5”* |
| *30 to 40* | *170 to 180* | *5’4” to 5’6”* |
| *40 to 50* | *175 to 185* | *5’6” to 6’0”* |
| *Above 50* | *Not Eligible* | *Not Eligible* |

*Prompt the user for the name, gender, age, weight with units (units could be kgs or lbs, for example , user could enter 90 kgs or 70 lbs) and height with units (units could be in cm or inches, for example , user could enter 5.5 inches or 165 cm).*

*If the user enters in pounds and inches cross check against the table and display whether he is eligible or not. If the user enters in kgs and cms convert the values and cross check against the table and display whether he is eligible or not. If a women candidate tries to apply, display a message that they are not eligible for the post.*

**Recommended duration:** *45 mins*

**Solution Guidance (if applicable):**  *1 kg equals 2.2 pounds, 1 Inch = 2.54 cm.*

**Lab Exercise No:** 50

**Exercise Objective(s):** *using non-static inner class*

**Exercise:**

*Write a java program which constructs a login name by using the inputs supplied by the user which are Full Name (Ex. R.Anand, Shanti Mohan) and the Account Number (5533881763. The rules are based on the following.*

* *First 4 characters of the login name should be alternate 4 characters of the full name including the initials starting from the beginning.*
* *Last 4 characters of the login name should be alternate 4 digits in the account number starting from the end.*
* *Whitespace is not allowed.*
* *First letter should be in uppercase and rest all should be in lower case*
* *No special character is allowed.*

**Recommended duration:** *45 mins*

**Solution Guidance (if applicable):**  *Build a character array and then use appropriate String constructor to build the string.*

**JAVA**

**Useful classes**

**Lab Exercise No:** 51

**Exercise Objective(s):** *using Calendar and Date*

**Exercise:**

* *A library has  books and each book has a number . Members can borrow only one book from the library and they must return the book within a week's time. If the return date is beyond 7 days, Rs.10 is charged as fine for each extra day. If the return date is beyond a month then Rs.50 is charged for each subsequent day after a month. For every subsequent month, 50 \* number of months is added for each extra day.*

*At the start of every month a bill is produced for all members with a token amount of Rs.100 to mark the continuance of the membership. In cases where the fine is applicable, the fine amount is also added.  
Write a java program to implement this application. The code must be able to take inputs to create books, members, dates (issue and return) and print the bill at the end of every month.*

**Recommended duration:** *1 hour*

**Solution Guidance (if applicable):**

**Lab Exercise No:** 52

**Exercise Objective(s):** *using Calendar and Date*

**Exercise:**

* *User enters the date in the form 7 July 2012. Display the colors based on the week it falls on*

*Mon is white*

*Tue is red*

*Wed is green*

*Thru is yellow*

*Fri is pink*

*Sat and Sun are not acceptable values.*

**Recommended duration:** *15 mins*

**Solution Guidance (if applicable):**

**Lab Exercise No:** 53

**Exercise Objective(s):** *using Calendar and Date*

**Exercise:**

*Write a java program to display the silver jubilee, golden jubilee, diamond jubilee celebration dates of a movie whose release date will be entered by the user. Assume that the movie will run successfully.(Silver Jubilee 25, Golden Jubilee 50, Diamond Jubilee 60, Platinum Jubilee 75).Note that if these dates fall on a Sunday or any public holiday then the date must be moved to next day.*

**Recommended duration:** *30 mins*

**Solution Guidance (if applicable):**

**Lab Exercise No:** 54

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

**Exercise:**

* *Write a java program to do the following.*

1. *Generate a random number between 1 and 100 and display the floor value and ceil value of the same.*
2. *Display the cube roots of the list of prime numbers till 100.*

**Recommended duration:** *30 mins*

**Solution Guidance (if applicable):**

**Lab Exercise No:** 55

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

**Exercise:**

*Program gets the input in the form of comma separated numbers, for example* ***"23,44,345.8“.***

*Write code that sums up the numbers and displays the result.*

**Recommended duration:** *30 mins*

**Solution Guidance (if applicable):**

**JAVA**

**CollectionAndGeneric**

**Lab Exercise No:** 56

**Exercise Objective(s):** ArrayList

**Exercise:**

* *Coordinator adds the names of participants who wish to participate in extempore. He also removes if participants decides otherwise or if they don’t meet the required criteria.*
* *An list is sorted and split as list of 5 participants and a seminar room number is allocated. This information is maintained as another list. Finally the application must display list as :*

*Group 1: seminar room*

*participants name*

*Group 2: seminar room*

*participants name*

*and so on*

**Recommended duration:** *45 mins*

**Solution Guidance (if applicable):**  *Use the ArrayList and Arrays class*

**Lab Exercise No:** 57

**Exercise Objective(s):** *Using Vector*

**Exercise:**

* *Create a Vector object that should be able to hold any type of object : Student or Teacher or HOD. Write a java code that creates these objects and inserts them into the list. Make sure that toString() is overridden in all the classes. Print out the list that displays the string representation of the object. It should also print the object type such as Student, Teacher or HOD.*

**Recommended duration:** *30 mins*

**Solution Guidance (if applicable):**

**Lab Exercise No:** 58

**Exercise Objective(s):** *Using Stack*

**Exercise:**

* *Post fix expression for ((2+3)\*8+5+3)\*6 is 6 5 2 3 + 8 \* + 3 + .*
* *Assuming that Post fix expression is given, find the result of the expression using Stack.*

*Hint:*

1. *Read the postfix expression from left to right character by character*
2. *If the input is an operand then push it onto the stack.*
3. *If the input is an operator then pop the two stack tops and perform the operation between the popped operands and push the result back into the stack.*
4. *Repeat the steps until we reach the end of the input.*

**Recommended duration:** *30 mins*

**Solution Guidance (if applicable):**

**Lab Exercise No:** 59

**Exercise Objective(s):** *Using Stack*

**Exercise:**

* *Post fix expression for ((2+3)\*8+5+3)\*6 is 6 5 2 3 + 8 \* + 3 + .*
* *Assuming that Post fix expression is given, find the result of the expression using Stack.*

*Hint:*

1. *Read the postfix expression from left to right character by character*
2. *If the input is an operand then push it onto the stack.*
3. *If the input is an operator then pop the two stack tops and perform the operation between the popped operands and push the result back into the stack.*
4. *Repeat the steps until we reach the end of the input.*

**Recommended duration:** *30 mins*

**Solution Guidance (if applicable):**

**Lab Exercise No:** 60

**Exercise Objective(s):** *Using Queue*

**Exercise:**

* *Implement a railway ticket counter scenario where there are two queues- one general and one for senior citizen.*
* *Tickets are issued such that for every one person in senior citizen queue, two persons in general queue are processed.*
* *Write a program that takes input for 6 people who come at various points and print the list of people in the order of their processing sequence.*

**Recommended duration:** *30 mins*

**Solution Guidance (if applicable):**

**Lab Exercise No:** 61

**Exercise Objective(s):** *Using Queue*

**Exercise:**

* *Use LinkedList to store list of Score objects (name, score) that will be entered by the user. Make sure they are arranged in the descending order of the scores. Display the linked list in the order of the rank of the student.*

**Recommended duration:** *30 mins*

**Solution Guidance (if applicable):**

**Lab Exercise No:** 62

**Exercise Objective(s):** *Using Set*

**Exercise:**

*Given an array of employee ids who were listed as outstanding for last 2 years.*

*Say : {1,2,6,3,4,5,6,7,9,4}*

*Write code picks the employees who are listed outstanding for 2 consecutive years.*

**Recommended duration:** *20 mins*

**Solution Guidance (if applicable):**

**Lab Exercise No:** 63

**Exercise Objective(s):** *Using Set*

**Exercise:**

*10 volunteers are needed for Showcase of New Product. Write a program that will accept employees ids who will volunteer for this. Make sure that the ids are not duplicates. Display the ids in a sorted order.*

**Recommended duration:** *20 mins*

**Lab Exercise No:** 64

**Exercise Objective(s):** *Using Map*

**Exercise:**

* *Write a class representing thesaurus that has many synonyms for a single word mapped. User can use this to search meaning of the words they want.*

**Recommended duration:** *20 mins*

**Solution Guidance (if applicable):**

**Lab Exercise No:** 65

**Exercise Objective(s):** *Using Map*

**Exercise:**

*Write a program to implement a telephone directory. Provide facilities to add, delete and search the telephone directory*

**Recommended duration:** *30 mins*

**Solution Guidance (if applicable):**

**Lab Exercise No:** 66

**Exercise Objective(s):** *Using Map*

**Exercise:**

* *A shop has a list of product code , description and price. Some prices are listed in terms of kg and others are listed in terms of dozens. Customers buys the different products in different quantities. The application must display a bill with the product code , description , quantity and price per unit and total price.*

**Recommended duration:** *45 mins*

**Solution Guidance (if applicable):**

**JAVA**

**IO**

**Lab Exercise No:** 67

**Exercise Objective(s):** *Using File and Directory*

**Exercise:**

* *Write a program that creates a new file called BatchMates.txt and store it in a directory named Batch. Also list the files or subdirectories present in the drive/directory where the directory Batch exists, stating if it is a file or directory.*

**Recommended duration:** *30 mins*

**Solution Guidance (if applicable):**

**Lab Exercise No:** 68

**Exercise Objective(s):** *Using Reader and Writer*

**Exercise:**

*Accepts 3 student’s roll no, name and grade and write each student details in a separate line in a file named student.csv. The details must be separated by comma.*

**Recommended duration:** *30 mins*

**Solution Guidance (if applicable):**

**Lab Exercise No:** 69

**Exercise Objective(s):** *Using Reader and Writer*

**Exercise:**

* *Ram wrote a Java code to create a text file that will store many file paths accessible by a large application with its timestamp. This code went live and later it was found that the file had the required data but instead of a new line separator between each network path, it had \n.*
* *Path.txt*
* *2012-1-30 T 10:45 UTC - E:\tomat \n 2012-1-30 T 12:45 UTC F:\ Data \n 2012-2-30 T 2:45 UTC - E:\MySQL*
* *Write java code that reads and replaces \n by new line character and write it to the file. Note that since the data in the file is very large , reading string data at one stroke may lead to overflow problems.*

**Recommended duration:** *30 mins*

**Solution Guidance (if applicable):** *you may need to read and unread characters*

**Lab Exercise No:** 70 (Home Work)

**Exercise Objective(s):** *Using byte stream*

**Exercise:**

* *Write a program using byte stream that is copies the content of one file into another file.*

**Recommended duration:** *45 mins*

**Solution Guidance (if applicable):** *you may need to read and unread characters*

**Lab Exercise No:** 71

**Exercise Objective(s):** *Using PrintStream*

**Exercise:**

* *Create a file called product list. Prompt the user to enter the name, price and manufacturing date. Save formatted data in the product list file so that name of the product should not exceed 10 characters, price is a double saved with precision as 2 chars after decimal such as 100.00 and date in format 05/29/06.*

**Recommended duration:** *45 mins*

**Solution Guidance (if applicable):** *you may need to read and unread characters*

**Lab Exercise No:** 72

**Exercise Objective(s):** *Using Serilaization*

**Exercise:**

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

**Recommended duration:** *30 mins*

**Solution Guidance (if applicable):** *you may need to read and unread characters*

**JAVA**

**XML**

**Lab Exercise No:** 73 (Home work)

**Exercise Objective(s):** *XML*

**Exercise:**

*Create a garden.xml for storing the information about tree, plants, and*

*herbs grown in a botanical garden. The following information should be stored about all the greens(trees, plants and herbs). Information about whether it is a trees, plants and herbs is given as attributes.*

* *Botanical name*
* *Age*
* *Suitable climate*
* *Medicinal use*

*Comment where ever necessary.*

*Verify well-formed-ness of the document.*

**Recommended duration:** *30 mins*

**Solution Guidance (if applicable):**

**Lab Exercise No:** 74

**Exercise Objective(s):** *XML*

**Exercise:**

* *Given:*

<?xml version="1.0" encoding='UTF-8'?>

<persons>

<person id="1234">

<name>Gayathri Sardar</name>

<address>C/O Vikki Sardar <br/>

172 Veera Apts., MG Road

<city>New Delhi</city>

<state>New Delhi</state>

</address>

</person>

<person id="2345">

<name>Bobby Mahajan

</name>

<address>C/O Deepak Mahajan <br/>

B501 &quot; Ashwini&quot; Apts., MG Road

<city>New Delhi</city>

<state>New Delhi</state>

</address>

</person>

<person id=“5678">

<name>Saranya Hari</name>

<address>W/O Hari Shankar<br/>

5/7 Elite Towers, LB Road

<city>Chennai</city>

<state>Tamil Nadu</state>

</address>

</person>

</persons>

*Write a program to print the names of the person who are from New Delhi.*

**Recommended duration:** *1 hour*

**Solution Guidance (if applicable):**

**JAVA**

**JDBC**

**Lab Exercise No:** 75

**Exercise Objective(s):** *Using JDBC, native drivers*

**Exercise:**

* *Create a table called Training in MYSQL with the following fields,*
  + *Sap\_ID*
  + *Employee\_name*
  + *Stream*
  + *Percentage*
* *Enter some data into the table without Percentage.*

1. *Write Java code using JDBC driver to display records.*
2. *Write Java code using native driver to display records one by one. Each time the record is displayed , prompt user to enter the Percentage and update the record.*

**Recommended duration:** *1 hrs*

**Solution Guidance (if applicable):**

**Lab Exercise No:** 76

**Exercise Objective(s):** *Using JDBC, large objects*

**Exercise:**

*Write a menu driven code to search, insert, update and delete from the Training table. Use advanced result set to do these.*

**Recommended duration:** *1 hr*

**Solution Guidance (if applicable):**

**Lab Exercise No:** 77

**Exercise Objective(s):** *Using JDBC, stored procedures*

**Exercise:**

* *Update the Training table to include photos also.*
* *Write java code that allows uploading photos to the database, for existing records in the table.*

**Recommended duration:** *1 hr*

**Solution Guidance (if applicable):**

**Lab Exercise No:** 78

**Exercise:**

* *Create a table called Account with AccNo, AccName, AccType and Balance. AccNo is the primary key.*
* *Write a stored procedure that returns the new AccNo which is computed as last entered AccNo +1. If the record is new, then AccNo is 1111111.*
* *Write java program , to allows insertion into the Account table.*

**Recommended duration:** *1 hr*

**JAVA**

**HTML**

**Lab Exercise No:** 79

**Exercise Objective(s):** *Using HTML tags*

**Exercise:**

* *Create a web page for a software company using HTML, which will contain two panes (Left pane and Right pane). In the left pane display few links like* 
  + *Home*
  + *About us*
  + *Clients*
  + *Careers*
  + *Contact us*
* *On clicking each of these links, display the content of these links in the right pane.*
* *When the user clicks on the Home link, display an image of the company with the company name in big letters. When the user clicks on the about us link, display some details about the company. When the user clicks on the Clients link, display the name of the few clients in a bulleted manner. When the user clicks on the Careers link, in the right pane display a user registration form for any openings in the company. The user registration form should get the following details from the user.*
* *Name, Date of birth, Gender(Radio button), Email id, Phone no, Educational Qualification(Drop down box).,Technology known(Check boxes), Submit button, Reset button*
* *When the user clicks on the Contact us link, display the location and contact details of the company across various countries in a table format.*

**Recommended duration:** *1 hrs*

**Solution Guidance (if applicable):**

**JAVA**

**JavaScript language fundamentals**

**Lab Exercise No:** 80

**Exercise Objective(s):** *Using alert and prompt*

**Exercise:**

* *Prompt the user to enter his/her full name. Form initials using the first characters of the first and last name. Popup the initials.*

*Hint :* ***String prompt(question,defaultanswer)***

* + *If user clicks* ***OK*** *it returns the string entered by the user*
  + *If user clicks* ***CANCEL*** *it returns* ***NULL***

**Recommended duration:** *20 mins*

**Solution Guidance (if applicable):**

**Lab Exercise No:** 81

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

**Exercise:**

* *Write a java script function to get a random number between 0 and 9?*

**Recommended duration:** *15 min*

**Solution Guidance (if applicable):**

**Lab Exercise No:** 82

**Exercise Objective(s):** *Using Date*

**Exercise:**

* *The electricity bill has to be paid before the 15th of every month. Write java script code that will prompt the user to enter the bill number and credit card number (assume that the site has secure payment gateway). If the date user pays the bill beyond 15th but before end of the month, then a fine of Rs. 50 is added for each delayed day. Compute the final bill and pop up the result to the user.*

**Recommended duration:** *30 min*

**Solution Guidance (if applicable):**

**Lab Exercise No:** 83

**Exercise Objective(s):** *Using DOM*

**Exercise:**

* *Display a page with one input box asking user to enter a number. On losing focus from the input box, compute factorial of the number. Display the result in red in a fresh page When the user closes the window it says “Bye”.*

**Recommended duration:** *30 min*

**Solution Guidance (if applicable):** *Use getElementsByTagName()and write() methods*

**Lab Exercise No:** 84

**Exercise Objective(s):** *Using form validation*

**Exercise:**

*In the web page that you created for a software company, there is a user registration form which will get the users details. On click of the submit button, validate the fields like Name, Date of birth, Email id, Phone no should not be empty. Also all the other fields value should be filled. And on clicking the reset button, clear the form.*

**Recommended duration:** *45 min*

**Solution Guidance (if applicable):**

**Lab Exercise No:** 85

***Exercise:***

*In the web page that you created for a software company, there is a user registration form which will get the users details. On click of the submit button, validate the fields like Name, Date of birth, Email id, Phone no. And on clicking the reset button, clear the form.*

**Recommended duration: 45 Min**

**JAVA**

**CSS**

**Lab Exercise No:** 86

**Exercise Objective(s):** *Using CSS*

**Exercise:**

*In the web page that you created for a software company, apply different styles using CSS based on following.*

* 1. *All the links should be underlined and with a blue color.*
  2. *All the headings should be in bold, uppercase, center aligned, red color and unique font from the text.*
  3. *All the labels in the form should be in times of roman and italics.*
  4. *Choose a appropriate color for the entire background for the web page.*

**Recommended duration:** *45 mins*

**Solution Guidance (if applicable):**

**JAVA**

**Servlet Basics**

**Lab Exercise No:** 87

**Exercise Objective(s):** *Using basic servlet methods*

**Exercise:**

*Aspire Height’s home page needs to have a new look after every ‘n’ number of visits. Create a servlet that initializes a variable in the servlet’s first method and increments the variable in the service method till the variable reaches the value 10. Once the value reaches 10, the servlet should be closed. Print the value during each increment and the method name in which it is incremented*

**Recommended duration:** *30 mins*

**Solution Guidance (if applicable):**

**Lab Exercise No:** 88

**Exercise Objective(s):** *Reading data form controls and using AJAX to update part of the page*

**Exercise:**

* *Implement a basic calculator application by getting the two operands and one operator from the form in a textbox. Also have a button called result and when the same is clicked display the result. Only the result part of the page must change*

**Recommended duration:** *1 hr*

**Solution Guidance (if applicable):** *use Ajax*

**Lab Exercise No:** 89

**Exercise Objective(s):** *Reading data form controls*

**Exercise:**

*Provide server side validation for the data entered in the user registration form (Name, Date of birth, Gender(Radio button), Email id, Phone no, Educational Qualification(Drop down box).,Technology known(Check boxes), Submit button, Reset button) that was created in HTML session.*

**Recommended duration:** *30 mins*

**Solution Guidance (if applicable):**

**JAVA**

**Servlet Initialization**

**Lab Exercise No:** 90

**Exercise Objective(s):** *Using config parameters*

**Exercise:**

*Create a web application with a servlet that displays user name and password. Validate the user name and password using the information provided in the config parameters . If the user name or password is invalid displays user name and password again till maximum X times where X is specified as context parameter.*

**Recommended duration:** *1 hour*

**Solution Guidance (if applicable):** *Implement using Annotation and web.xml.*

**Lab Exercise No:** 91

**Exercise Objective(s):** *Using config parameters*

**Exercise:**

*Create a Servlet that acts as a filter . This servlet does not allow IPs that are forbidden which are specified as config parameters. If the remote address represents the IP specified in the config parameters, display a forbidden access page. Otherwise, a success access page message is displayed.*

**Recommended duration:** *45 mins*

**Solution Guidance (if applicable):** *HttpServletRequest ‘s getRemoteAddr*

**JAVA**

**ServletCommunication**

**Lab Exercise No:** 92

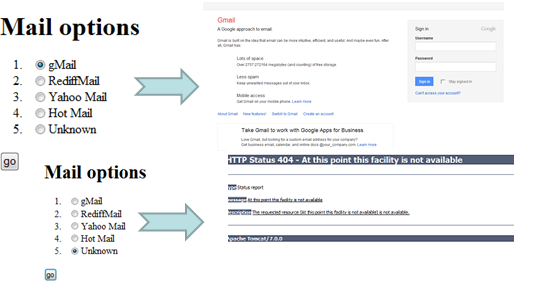
**Exercise Objective(s):** *Redirecting*

**Exercise:**

*Write a web application that does the following:*

* *HTML page displays email options of which one is unknown.*

*If unknown option is an 404 error page is displayed.*

**

**Recommended duration:** *30mins*

**Solution Guidance (if applicable):**

**Lab Exercise No:** 93

**Exercise Objective(s):** *forward and include*

**Exercise:**

*User can choose to go to flowers page or cards page from the home page and choose the items displayed in the page. From cards (or flowers) page he/she can go to flowers(or cards) page and bill page. In the bill page amount for the items selected in the cards page and flowers page is listed. In case neither flower nor card is selected, then an error message is displayed along with the home page.*

**Recommended duration:** *30 mins*

**Solution Guidance (if applicable):**

**Lab Exercise No:** 94

**Exercise Objective(s):** *forward and include*

**Exercise:**

*Add the filter servlet that you created in the last session to the above application. All the request would now pass through this filter servlet. If the remote address represents the IP specified in the config parameters, display a forbidden access page. Otherwise, any page requested by the user is displayed.*

**Recommended duration:** *45 mins*

**Solution Guidance (if applicable):**

**Lab Exercise No:** 95

**Exercise Objective(s):** *reading from a passive resource*

**Exercise:**

* *Display the product catalog that is provided in the text file with the prices in a web page using a servlet.*
* *.*

**Recommended duration:** *15 mins*

**Solution Guidance (if applicable):**

**JAVA**

**Connecting to database**

**Lab Exercise No:** 96

**Exercise Objective(s):** *connecting to database from a web page*

**Exercise:**

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

* *Sap\_ID*
* *Stud\_name*
* *Stream*
* *Percentage*

*Insert all your batch mates’ details and create a web page that displays the details of the table.*

*The config information about the database should be supplied through the web.xml.*

**Recommended duration:** *20 mins*

**Solution Guidance (if applicable):**

**Lab Exercise No:** 97

**Exercise Objective(s):** *connecting to database from a web page*

**Exercise:**

*Create a page with the list of energy drinks like boost, horlicks, complain, bournvita, and pediasure available in a shop. Let the user select any number of varieties he wants and provide a text box for each variety to specify the quantity. On completion of the selection instruct the user to press the Submit button. Maintain a database with the following fields like Item name, Quantity available, Price, Expiry date etc. On clicking the submit button, display the details to the user for the items that he has selected to continue shopping.*

**Recommended duration:** *40 mins*

**Solution Guidance (if applicable):**

**JAVA**

**Session Handling**

**Lab Exercise No:** 98

**Exercise Objective(s):** *Using hidden fields*

**Exercise:**

* *Rewrite the shopping card application written in the previous example using hidden field to maintain state.*

**Recommended duration:** *30 mins*

**Solution Guidance (if applicable):**

**Lab Exercise No:** 99

**Exercise Objective(s):** *Using HttpSession*

**Exercise:**

*A table contains 10 yes/no questions. When user takes exam , 3 questions are randomly picked from the table and displayed. Maximum of 3 mins is given for the exam. Each question is displayed in a separate page. If user can use next and prev buttons to navigate. The first question page should have prev button and last question page should not have next button. There should be a finish button in every question page. If user has answered a question, then the user’s answer must be shown when the question page is displayed. When user clicks on the finish button or if the time is beyond 3 mins, the result page with marks must be displayed.*

*Write a servlet based web application to do this.*

**Recommended duration:** *1 hr**45 mins*

**Solution Guidance (if applicable):**

**JAVA**

**Listeners**

**Lab Exercise No:** 100 (Homework)

**Exercise Objective(s):** *Using context listeners*

**Exercise:**

* *All the attributes that are added to the context must also be maintained in the database table with the name and value. Similarly they must be removed when the attribute is removed.*

**Solution Guidance (if applicable):**

**Lab Exercise No:** 101

**Exercise Objective(s):** *Using session listeners*

**Exercise:**

* *For the exam web application that you created, if user accidently closes the browser, then he must be able to resume the exam. To provide this facility, when ever any attribute is added or updated in the session, it must be added to the database too. Similarly when ever it is removed, it must be removed. Apart from that status of the exam must also be maintained.*

*On requesting for exam page, if the exam status is incomplete, then the last question that user attempted must be shown. User must be able to resume seamlessly.*

**Recommended duration:** *1 hr**30 mins*

**Solution Guidance (if applicable):**

**JAVA**

**Introduction to JSP**

**Lab Exercise No:** 102

**Exercise Objective(s):**  *Form data handling using JSP*

**Exercise:**

*C Create a JSP that displays 2 text boxes. Accept two numbers and print the addition, subtraction, multiplication and division of those numbers.*

**Recommended duration:** *30 mins*

**Solution Guidance (if applicable):**

**Lab Exercise No:** 103

**Exercise Objective(s):**  *Form data handling using JSP*

**Exercise:**

* *Display a JSP registration form that requests the user to enter user’s name, email id, date of birth, address, phone number. On submitting the form, a servlet must validate the data. In case the data is invalid, the JSP page must be displayed again indicating the error fields. This time the registration form must be prefilled with the data that user has entered previously. If the data is valid then*

1. *If a record already exists in the table, ask the user if the records have to be updated. If user click yes, the display the JSP registration page with successful updation after updating the database. If user clicks no, then fetch the record from the database and display the record in the JSP registration page.*
2. *If the record does not exist then insert the data and show the page with the relevant data and successful insertion message.*

*The database parameters must be in the web.config*

**Recommended duration:** *1 hr 30 mins*

**Solution Guidance (if applicable):**

**Lab Exercise No:** 104

**Exercise Objective(s):**  *Handling error*

**Exercise:**

* *In the previous exercise, in case an error occurs, on account of a non-number being entered or if an attempt to divide by 0 occurs the page should automatically be redirected to error page.*

**Recommended duration:** *15 mins*

**Solution Guidance (if applicable):**

**Lab Exercise No:** 105 (Home Work)

**Exercise Objective(s):**  *using multiple JSP*

**Exercise:**

* *There are 2 pages- Images and Cards. The user can select a card style (border thickness, border color, font size and color) and add personal message in the Cards page. He /She then can select image from the Images page. Finally on clicking “Preview” a page with the image inside the boundary as specified by the card style and message in the font and color specified is displayed. One final submit, email details of the user to whom this has to be sent is prompted. On clicking “Send” , “Sent successfully” page is displayed.*

**Recommended duration:** *45 mins*

**Solution Guidance (if applicable):**

**JAVA**

**JSP Action tags/Standard Actions**

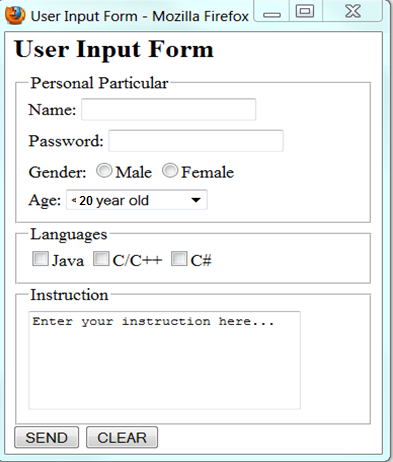
**Lab Exercise No:** 106

**Exercise Objective(s):**  *using standard actions*

**Exercise:**

*Create a form as shown.*

*Create a bean for the input form and when “Send” button is clicked, it should display the user values in a table in the next page*

****

**Recommended duration:** *1 hr*

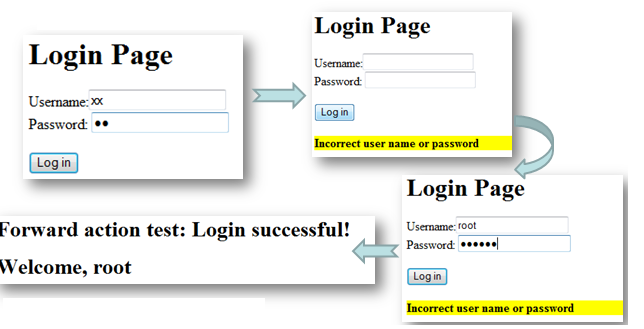
**Solution Guidance (if applicable):**

**Lab Exercise No:** 107

**Exercise Objective(s):**  *using standard actions*

**Exercise:**

* *User enters login and password which is validated. If invalid values are provided, login page is displayed with error message, otherwise welcome page is displayed.*

**

**Recommended duration:** *1 hr*

**Solution Guidance (if applicable):**

**Lab Exercise No:** 108

**Exercise Objective(s):**  *using JSP scriplets, standard actions, directives*

**Exercise:**

* *Stamp collectors can use this site to post details of stamp that they would like to auction. Interested collectors can bid to purchase the stamp. Each item should be available for 3 days. At the end of three days person whose bid amount is highest, gets the stamp.*

**Recommended duration:** *1 hour*

**Solution Guidance (if applicable):** *change the system date to test the application.*

**JAVA**

**Custom Tags**

**Lab Exercise No:** 109

**Exercise Objective(s):**  *using simple custom tag without body*

**Exercise:**

* *Write a custom tag called Hello Tag which will print "Hello! Welcome to the world of custom tags" wherever <custom:hello/> would be used.*
* *What happens if the tag with invoked with a body?*

**Recommended duration:** *30 mins*

**Solution Guidance (if applicable):**

**Lab Exercise No:** 110

**Exercise Objective(s):**  *using simple custom tag*

**Exercise:**

* *Write a custom tag called TodayTag which will print today’s date wherever <custom:today> would be used. The date should be printed in the format that is specified as an attribute in the today tag. Implement using Simple Custom tags.*

**Recommended duration:** *30 mins*

**Solution Guidance (if applicable):**

**Lab Exercise No:** 111

**Exercise Objective(s):**  *using simple custom tag*

**Exercise:**

* *Write a custom tag called TodayTag which will print today’s date wherever <custom:today> would be used. The date should be printed in the format that is specified as an attribute in the today tag. Implement using Simple Custom tags.*

**Recommended duration:** *30 mins*

**Solution Guidance (if applicable):**

**Lab Exercise No:** 112

**Exercise Objective(s):**  *using classic custom tag*

**Exercise:**

* *Write a custom tag called ChangeCaseTag which will convert the body content of the tag to either upper case or lower case based on the case specified in the attribute of the <custom:changecase> tag. Implement using Simple Custom tags.*

*(30 Mins)*

**Recommended duration:** *30 mins*

**Solution Guidance (if applicable):**

**Lab Exercise No:** 113

**Exercise Objective(s):**  *using classic custom tag*

**Exercise:**

* *Write a custom tag called EmptyTag which will print "This is an empty tag!" wherever <custom:empty> would be used. The tag should print this line and rest of the body content should be ignored. Implement using Classic Custom tags.*

**Recommended duration:** *45 mins*

**Solution Guidance (if applicable):**

**Lab Exercise No:** 114

**Exercise Objective(s)**

**Exercise:**

* *Write a custom tag called Loop Tag which will print the body of the text ‘n’ times where ‘n’ is specified as the attribute of the tag. Implement using Classic Custom tags.*

**Recommended duration: 45 min**