

Jawaharlal Nehru Engineering College

Laboratory Manual

Core Java

for

S.Y. (M.C.A) Students

26, June 2019 – Rev 00 – MCA – ISO 9001 Tech Document

✚ Author JNEC, Aurangabad

FORWARD

It is my great pleasure to present this laboratory manual for SY (MCA) students for the subject of CJ (Core Java). Subject name itself is expecting programming skills that are needed as basis of almost all programming language.

As a student, many of you may be wondering with some of the questions in your mind regarding the subject and exactly what has been tried is to answer through this manual.

As you may be aware that MGM has already been awarded with ISO 9001 certification and it is our endure to technically equip our students taking the advantage of the procedural aspects of ISO 9001 Certification.

Faculty members are also advised that covering these aspects in initial stage itself, will greatly relived them in future as much of the load will be taken care by the enthusiasm energies of the students once they are conceptually clear.

Dr. H.H.SHINDE
Principal

LABORATORY MANUAL CONTENTS

This manual is intended for the students SY MCA for the subject of CJ (Core Java). Subject name itself is expecting what are basic and the advance skills that might be requires as a basis of a programming language. In the subject of this manual typically contains practical/Lab Sessions we have dealt with Core Java programming language. (data types, operators, arrays, classes and files.

Students are advised to thoroughly go through this manual rather than only topics mentioned are the key to understanding and conceptual visualization of theoretical aspects covered in the books.

Good Luck for your Enjoyable Laboratory Sessions

Prof. V.S.Agrawal

Dr.S.S.Deshmukh

Asst.Prof

HOD MCA

SUBJECT INDEX:

- 1) a) Program on creation of classes & functions.
b) Program using constructor/function overloading
c) Program on passing object as a parameter to function

- 2) a) Program using interface and inheritance covering domain like educational institute/bank etc
b) Program to perform operations on Array & String.

- 3) Program using packages to demonstrate scope of access specifiers.

- 4) a) Program on Exception handling covering (try, catch, throw, throws, finally)
b) Program to create your own exception class.

- 5) Program to demonstrate collection

- 6) a) Program to create multiple threads doing different tasks.
b) Program to demonstrate thread priorities.

- 7) Program to perform database operations (select/insert/update/delete)

- 8) Program using IO streams

- 9) Program to demonstrate applet.

- 10) a) Program to perform addition, subtraction, multiplication, division using AWT controls.
b) Program to perform event handling (awt/applet)

DOs and DON'T DOs in Laboratory:

1. Do not handle any equipment before reading the instructions/Instruction manuals
2. Read carefully the power ratings of the equipment before it is switched on whether ratings 230 V/50 Hz or 115V/60 Hz. For Indian equipments, the power ratings are normally 230V/50Hz. If you have equipment with 115/60 Hz ratings, do not insert power plug, as our normal supply is 230V/50 Hz, which will damage the equipment.
3. Observe type of sockets of equipment power to avoid mechanical damage
4. Do not forcefully place connectors to avoid the damage
5. Strictly observe the instructions given by the teacher/Lab Instructor

Instruction for Laboratory Teachers:

1. Submission related to whatever lab work has been completed should be done during the next lab session. The immediate arrangements for printouts related to submission on the day of practical assignments.
2. Students should be taught for taking the printouts under the observation of lab teacher.
3. The promptness of submission should be encouraged by way of marking and evaluation patterns that will benefit the sincere students.

WARMUP EXERCISES:

What is software and hardware?

What is a Java?

What are Java Versions?

What is a JDK?

What is a JRE?

Lab Exercises:

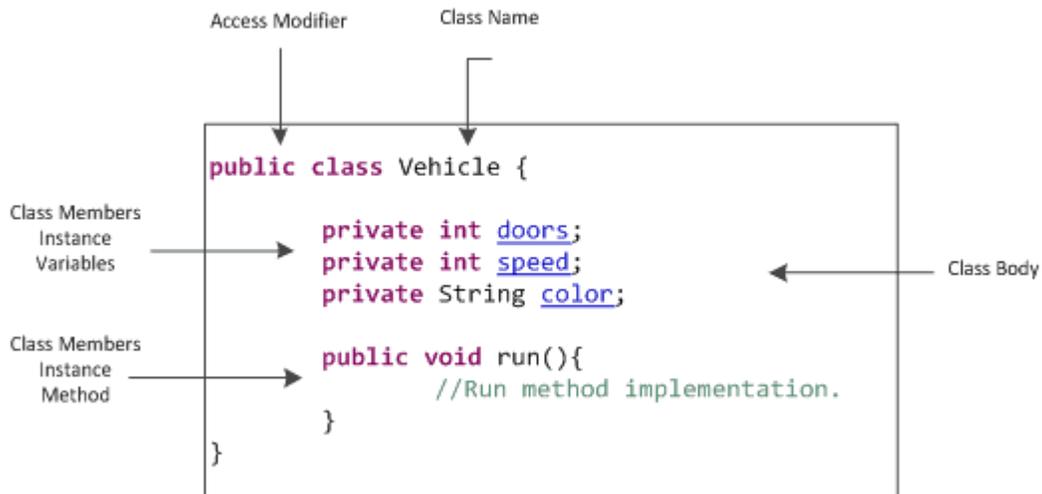
Exercise No1: (2 Hours) – 1 Practical

a) Program on creation of classes & functions.

Class: - It is collection of data members and member functions.

Declaration of Class:

A class is declared by use of the class keyword. The class body is enclosed between curly braces { and }. The data or variables, defined within a class are called instance variables. The code is contained within methods. Collectively, the methods and variables defined within a class are called members of the class.



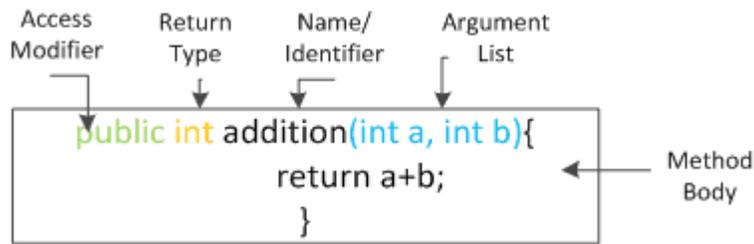
Declaration of Instance Variables :

Variables defined within a class are called instance variables because each instance of the class (that is, each object of the class) contains its own copy of these variables. Thus, the data for one object is separate and unique from the data for another. An instance variable can be declared public or private or default (no modifier). When we do not want our variable's value to be changed out-side our class we should declare them private. public variables can be accessed and changed from outside of the class. We will have more information in OOP concept tutorial. The syntax is shown below.



Declaration of Methods:

A method is a program module that contains a series of statements that carry out a task. To execute a method, you invoke or call it from another method; the calling method makes a method call, which invokes the called method. Any class can contain an unlimited number of methods, and each method can be called an unlimited number of times. The syntax to declare method is given below.



Steps for program:

- 1) start the program
- 2) create class Box
- 3) declare instance variables width,height and depth
- 4) define method volume() which will calculate and display the volume of a box.
Formula:- $\text{volume} = \text{width} * \text{depth} * \text{height}$
- 5) define main() in which create the object of class Box.
- 6) initialize instance var of class using an object.
- 7) call the volume() function using object.
- 8) end of the program

Input:- h-6 w-2 d-4

Output:- Volume of box is 48.

b) Program using constructor overloading

Constructor:-

In Java, a constructor is a block of codes similar to the method. It is called when an instance of the class is created. At the time of calling constructor, memory for the object is allocated in the memory.

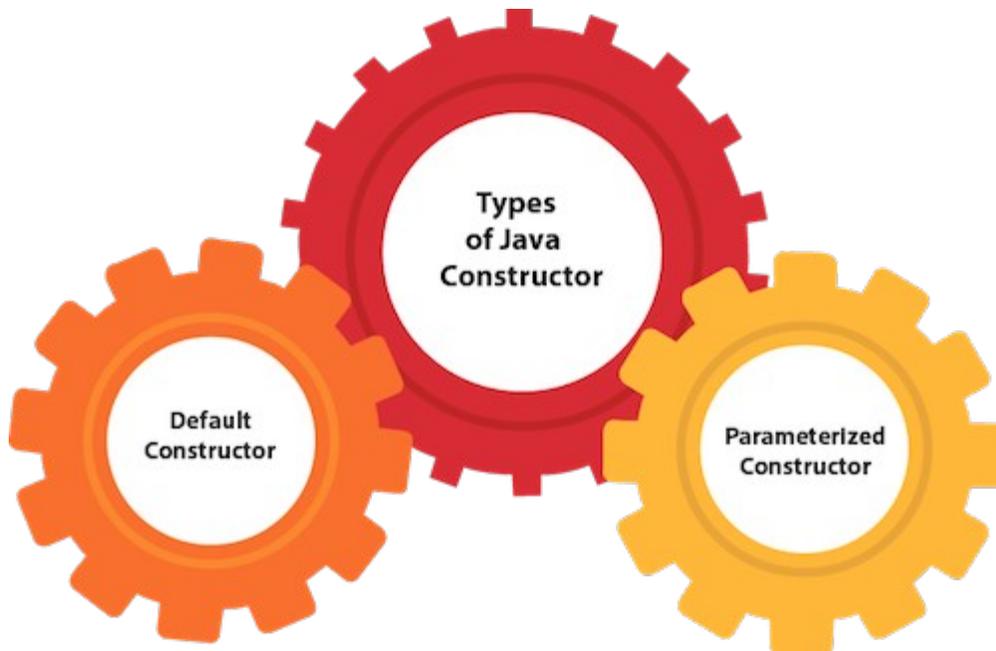
It is a special type of method which is used to initialize the object.

Every time an object is created using the new() keyword, at least one constructor is called.

Types of Java constructors

There are two types of constructors in Java:

1. Default constructor (no-arg constructor)
2. Parameterized constructor



Constructor Overloading in Java

In Java, a constructor is just like a method but without return type. It can also be overloaded like Java methods.

Constructor overloading in Java is a technique of having more than one constructor with different parameter lists. They are arranged in a way that each constructor performs a different task. They are differentiated by the compiler by the number of parameters in the list and their types.

Steps for program:

- 1) start the program
- 2) create the class Box
- 3) declare the instance var as width,height and depth.
- 4) define the default constructor(without argument)
- 5) define parameterized constructor(3 args as w,h & d)

- 6) define parameterized constructor(1 argument len in case of same values of w,h & d)
- 7) define the volume() method which will calculate and display the volume.
- 8) define main() function in which create the object by passing the arguments
- 9) call the volume() function
- 10) end of the program.

Input:- h-6 w-2 d-4

Output:- Volume of box is 48.

Input: if no input values passed then it will call default constructor

Output: Volume of box is 8 (value 2 is passed to all variables)

c) Program using function overloading

Method Overloading in Java

If a class has multiple methods having same name but different in parameters, it is known as **Method Overloading**.

If we have to perform only one operation, having same name of the methods increases the readability of the program.

Suppose you have to perform addition of the given numbers but there can be any number of arguments, if you write the method such as a(int,int) for two parameters, and b(int,int,int) for three parameters then it may be difficult for you as well as other programmers to understand the behavior of the method because its name differs.

Steps for program:

- 1) start of the program
- 2) create class Function
- 2) define method addition with 2 parameters as int
- 3) define method addition with 2 parameters as float
- 4) define method addition with 2 parameters as int & float
- 5) define main() function in which create the object of class Addition
- 6) call the method addition() method multiple times by passing different kind of parameters.

Input:- 3 4

Output: Addition is 7

d)Program on passing object as a parameter to function

As we know it is core concept that in Java there is always pass by value and not by pass by reference. So in this post we will focus on that how this concept get validated in case of passing primitive and passing reference to a method.

In case when a primitive type is passed to a method as argument then the value assigned to this primitive is get passed to the method and that value becomes local to that method, which means that any change to that value by the method would not change the value of primitive that you have passed to the method.

While in case of passing reference to a method again Java follow the same rule of pass by value now let understand how it happens.

As we know that a reference in Java holds the memory location of object created if it is assigned to that reference otherwise it is initiated as null. Now the point to remember here is that the value of the reference is the memory location of the assigned object, so whenever we pass the reference to any method as argument then we actually pass the memory location of that object which is assigned to that particular reference. This technically means that the target method has memory location of our created object and can access it to. So in case if target method access our object and make changes to any of property of it than we would encounter with the changed value in our original object.

Steps for program:

- 1) start of the program
- 2) create the class Box
- 3) declare the instance var as width,height and depth.
- 4) define the default constructor(without argument)
- 5) define parameterized constructor(3 args as w,h & d)
- 6) define the volume() method which will calculate the volume.
- 7) define method check() by passing object as a argument(Box b1,b2) in which compare the volume of both the boxes if found same then display “boxes are of same size” otherwise display “boxes are of different sizes”
- 8) end of the program

Input:- Enter height, width and depth of first box
2 3 4

Enter height, width and depth of second box
2 3 4

Output:- Both boxes are of same sizes

Input:- Enter height, width and depth of first box
2 3 5

Enter height, width and depth of second box
2 3 4

Output:- Both boxes are of different sizes

Exercise No 2: (2 Hours) – 1 Practical

a) Program using interface and inheritance covering domain like educational institute/bank etc

Inheritance in Java

Inheritance in Java is a mechanism in which one object acquires all the properties and behaviors of a parent object. It is an important part of OOPs (Object Oriented programming system).

The idea behind inheritance in Java is that you can create new classes that are built upon existing classes. When you inherit from an existing class, you can reuse methods and fields of the parent class. Moreover, you can add new methods and fields in your current class also.

Inheritance represents the **IS-A relationship** which is also known as a *parent-child* relationship.

The syntax of Java Inheritance

1. **class** Subclass-name **extends** Superclass-name
2. {
3. //methods and fields
4. }

Interface in Java

An **interface in java** is a blueprint of a class. It has static constants and abstract methods.

The interface in Java is *a mechanism to achieve abstraction*. There can be only abstract methods in the Java interface, not method body. It is used to achieve abstraction and multiple inheritance in Java.

In other words, you can say that interfaces can have abstract methods and variables. It cannot have a method body.

Steps:

- 1) start of the program
- 2) define interface Account which will contain 2 method deposit() & withdraw()
- 3) create class BankAccount with 2 instance var acno and acname, define method createAccount() which will ask user details & display acno with success message. Define displayinfo() method which will display acno,acname.
- 4) create class saving which extends BankAccount class and implements Account interface having bal as instance var.
Define deposit() method which will add the amt entered by user & display current balance.
Define method displayBalance()- which will call displayinfo() method to display acno and acname, and display current saving balance.
Define method withdraw() which will check amt>bal then display message as “take the permission from bank” otherwise deduct the amt and display the balance.
- 5) create class current extends BankAccount class and implements Account interface having bal as instance var.
Define deposit() method which will add the amt entered by user & display current balance.
Define method displayBalance()- which will call displayinfo() method to display acno and acname, and display current saving balance.
Define method withdraw() which will check bal<500 then calculate the penalty & display message as “balance after deducting the penalty is” or check amt>bal then display the message as “take the permission from bank” or deduct the amt and display the balance.
- 6) create the class BankDemo in which define main() function and ask the user to select the option saving or current.
Then ask the user to select the option such as createaccount, deposit, withdraw, displaybalance etc.

Input & Output:-

Enter option

- 1.saving
- 2.current

Say saving

- 1,create account
- 2.deposit
- 3.withdraw

4.display balance

Say 1

Enter customer name:- abc

Your acc no is 1

Balance is 500

Enter your choice

1.create account

2.deposit

3.withdraw

4.display balance

Say 4

Your balance amount is 500

b)

I)Program to perform operations on Array.

Normally, an array is a collection of similar type of elements that have a contiguous memory location.

Java array is an object which contains elements of a similar data type. It is a data structure where we store similar elements. We can store only a fixed set of elements in a Java array.

Array in java is index-based, the first element of the array is stored at the 0 index.

Steps:

- 1) start of the program
- 2) create 2 arrays
- 3) display 2 array before swapping
- 4) perform swapping of 2 arrays.
- 5) display 2 arrays after swapping
- 6) end of the program

Input:- enter first array elements 1 2 3

enter second array elements 4 5 6

Output:-

Before swapping

First array is 1 2 3

Second array is 4 5 6

After Swapping

First array is 4 5 6

Second array is 1 2 3

II)Program to perform operations on String

String :- It is collection of characters.

Steps:

- 1) start of the program
- 2) create string
- 3) use method charAt(index) to display char at particular location
- 4) use method toUpperCase() to display string in uppercase letters
- 5) use method toLowerCase() to display string in Lowercase letters
- 6) use method indexOf() to get particular location character.
- 7) use method lastIndexOf() to display last index of particular character
- 8) end of the program

Input:- Enter a string india

Output:-

Character at index 0 is i

Uppercase INDIA

Lowercase india

Indexof(i):- 0

Lastindexof(i) :- 3

Exercise No 3: (2 Hours) – 1 Practical

Program using packages to demonstrate scope of access specifiers.

Access Specifiers:- There are various access specifiers like public, private, protected, default.

By default default access specifier used.

Scope of access specifiers

Default:- within the package

Public:- anywhere inside & outside program

Private:- within class only

Protected:- within package, outside package if it is subclass.

Steps:

1) start of the program

2)

// same class

create class Demo in which declare 3 instance var as j-private, l-public, x-protected, m-no access specifier.

Try to access j – yes as it is private

Try to access l – yes as it is public

Try to access x – yes as it is protected

Try to access m – yes as it is default

3)

//same package child class

create one more class Check which extends class Demo in which

try to access j, it will not be accessed as it is private.

Try to access l, yes as it is public

Try to access x, yes as it is protected

Try to access m, yes as it is default

4)

//different package, simple class

create another package in which

Try to access i, No as it is private

Try to access l, yes as it is public

Try to access m, No as it is default

Try to access x, No as it is protected

5)

//different package child class

create another class in current package

Try to access i, No as it is private

Try to access l, Yes as it is public

Try to access m, Yes as it is default

Try to access x, as it is protected.

7) end of the program

Exercise No 4: (2 Hours) – 1 Practical

a)Program on Exception handling covering (try, catch, throw, throws, finally)

Exception:- Exception is run time error. There are various keywords used in exception handling like try, catch, throw, throws, finally etc

Try:- block of code in which exception may occur

Catch:- type of exception which may get thrown

Finally:- code to be executed whether or not exception occurs

Throw:- to throw exception

Throws:- whenever code don't want to handle the exception.

steps:

1) start of the program

2) create class Demo

3) define method read() which will 2 numbers.

If numbers are not numeric then it will throw InputMismatchException

4) define method div() which will calculate and display division of the two numbers.

If denominator is 0 then it will throw ArithmeticException

5) catch the both exception classes.

6) in finally block perform the addition of 2 numbers

7) end of the program.

Input:-

Enter 2 numbers

2

0

Output:-Dividend can not be zero.

Input:-

Enter 2 numbers

2

g

Output:-enter numeric data only

b)Program to create your own exception class.

Steps:

- 1) start of the program
- 2) create class MyException which extends class Exception
- 3) define constructor parameterized which will pass string to super class i.e. class Exception using keyword super.
- 4) create another class Demo in which instance variables are user and pass. Define method check() to validate username and password. If username is invalid then throw exception by passing comment "username is invalid" and if pass is invalid then throw the exception by passing comment "password is invalid"
- 5) end of the program.

Input:- enter userid and password

Abc

Def

Note:- actual userid is Abc and password is Def

Output:- password is invalid

Exercise No 5: (2 Hours) – 1 Practical
Program to demonstrate collection

Collection:- It is set of data or objects. It is different than array. Array provides only storage space whereas collection provides storage space as well as inbuilt functions to work on that data.

There are various classes and interfaces in case of collection.

HashTable:- Which works on hashing concept.

Data is always stored based on keys.

We can minimize searching time using HashTable.

Steps:

- 1) start of the program
- 2) create class Hash
- 3) define main() function in which create object of class HashTable say ht.
- 4) use put() method To insert data as <key,value> pair.
- 5) use clear() to make empty the hashtable
- 6) use remove() method to delete data at particular key.
- 7) use contains() method to search data
- 8) end of the program.

Input and Output:-

101 abc
102 def
103 pqr

Say remove 103

Now records would be
101 abc
102 def

Enter record to search :- 109

Data not found

Enter record to search :- 102

Data found

Exercise No 6: (2 Hours) – 1 Practical

a) Program to create multiple threads doing different tasks.

Thread:- It is part of process. We can execute multiple thread parallel provided with all thread jobs are independent. As if they are dependent we can't execute them in parallel.

Multithreading:- Running multiple threads simultaneously.
Each thread will have its own job to be executed.

Steps:

- 1) start of the program
- 2) create class A which extends Thread class
Define method run() in which write the logic(process) of first thread.
- 3) define another class B which extends Thread class
Define method run() in which write the logic of second thread.
- 4) define main() function in which create object of both the classes.
- 5) use start() method to activate both the run() methods.
- 6) display the output in run() method.
- 7) end of the program

Input:- start both the threads

Output:-

1
2
3
4
5
6

Here 1 2 3 were of first thread and 4 5 6 are of second thread.

b) Program to demonstrate thread priorities

Thread Priority:- There are 3 basic priorities.
1:-low 5:-medium 10:-high

Say 2 threads having different priority then the thread which has highest priority will execute first followed by second thread.

steps:

- 1) start of the program
- 2) create class Demo which extends Thread class
- 3) define method run() in which write the logic.
- 4) define main() function in which create 2 objects of same class
- 5) use method setPriority() to assign the priority to the threads.
- 6) use start() method to activate the threads.
- 7) end of the program.

Input

1 2 3 of first thread

4 5 6 of second thread

Set priority highest to second thread and lowest to first thread.

Run both the threads simultaneously.

Output:- 4 5 6 1 2 3

Exercise No 7: (2 Hours) – 1 Practical

Program to perform database operations (select/insert/update/delete)

Database:- We can store n number of tables inside database.

We can refer any relational database to store the data.

Steps:

- 1) start of the program
- 2) create class Demo
- 3) define main() function in which
- 4) define driver name
- 5) create the connection
- 6) generate statement.
- 7) use executeQuery() method to display the database records at front end.
- 8) use excuteUpdate() method to insert/delete/update the records to/from the database.
- 9) close stament and connection objects.
- 10) end of the program

Input and output:-

Displaying students records

1 a

2 b

3 c

4 d

Inserting data

5 e

Display

1 a

2 b

3 c

4 d

5 e

Update

5 p

Display

1 a 2 b 3 c 4 d 5 e

Exercise No 8 : (2 Hours) – 1 Practical
Program using IO streams

Before database we used to use file.
File is collection of data which is in unstructured format.

Disadvantage:- Hard to handle with data.

Steps:

//copying one file into another

- 1) start of the program
- 2) create class Demo
- 3) crate 2 objects of class File
- 4) open first file in read mode
- 5) open second file in write mode
- 6) apply the loop ,read and write from one file into another file
- 7) display the output as” file copied successfully”
- 8) end of the program

Input & output

Say file1.txt contains

Abcdef

When we execute the program it copies the first file contents into second file character by characters.

Output:- file copied successfully.

file1.txt contains

Abcdef

File2.txt contains

Abcdef

Exercise No 9 : (2 Hours) – 1 Practical

a) Program to demonstrate applet

Applet:- It is java program used to execute at client side. It uses resources from client system to minimize buffering in case of animations.videos.

Steps:

//to display string in applet

- 1) start of the program
- 2) create class Demo which extends class Applet
- 3) use paint() method to display string on front end
- 4) end of the program

Input:- using paint method display hello

Output:- hello

b) program using swing

Swing :- It is advance version of AWT. It is lightweight and has many more components . Swing can be used to create desktop applications. It has good look and feel.

steps:

// to display radio button name in textbox selected by end user

- 1) start of the program
- 2) create class Demo which extends JApplet class & implements ActionListener interface
- 3) crate 1 textfield, 2 radio buttons
- 4) add the objects to the front end using add() method.
- 5) define actionPerformed() method in which check which radio buton user has selected using getActionCommand() method & display the name of that radio button in the textbox.
- 6) end of the program.

Input:- select one from given three choices.

Output:- One

Exercise No 10 : (2 Hours) – 1 Practical

a)Program to display status of the checkboxes using applet i.e.true/false i.e.selected /deselected(Program to perform event handling (awt/applet

steps:

- 1) start of the program
- 2) create class Demo which extends Applet class and implements ItemListener interface
- 3) define init() method in which create the objects of Checkbox classes to generate checkboxes
- 4) add all the objects to the front end using add() method
- 5) use addItemListener() method for event handling
- 6) define method itemStateChanged() in which call repaint() method to refresh applet output
- 7) define paint() method in which display the string which will states the current status of the checkbox i.e. true/flase i.e.selected/deselectd.
- 8)end of the program.

Input:- select cricket and volleyball checkbox

Output:-

Cricket: true
Volleyball:true
Badminton:false

b)Program to perform addition, subtraction, multiplication, division using AWT controls.

Steps:

- 1) start of the program
- 2) create class Demo which will extends Applet class and implements ItemListener interface
- 3) define init() method in which create objects of textbox,radio buttons
- 4) add all the objects usibng add() method.
- 5) define method itemStateChanged() to check which radio buton end user has selected.
As per the selection perform the particular operation add/sub/mul/div .
- 6) display the answer into the texbox also by calling repaint() method.

- 7) define print() method to display the string as a output.
- 8) end of the program.

Input:-enter 2 numbers say 3, 4 and select addition

Output:- addition is 7

Input:-enter 2 numbers say 3, 4 and select multiplication

Output:- multiplication is 12

Input:-enter 2 numbers say 3, 4 and select division

Output:- division is 0

Quiz on the subject:

Quiz should be conducted on tips in the laboratory, recent trends and subject knowledge of the subject. The quiz questions should be formulated such that questions are normally are from the scope outside of the books. However twisted questions and self formulated questions by the faculty can be asked but correctness of it is necessarily to be thoroughly checked before the conduction of the quiz.

Conduction of Viva-Voce Examinations:

Teacher should oral exams of the students with full preparation. Normally, the objective questions with guess are to be avoided. To make it meaningful, the questions should be such that depth of the students in the subject is tested Oral examinations are to be conducted in co-cordial environment amongst the teachers taking the examination. Teachers taking such examinations should not have ill thoughts about each other and courtesies should be offered to each other in case of difference of opinion, which should be critically suppressed in front of the students.

Submission: Each student has to submit journal for this subject by completing all the assignments.

Evaluation and marking system:

Basic honesty in the evaluation and marking system is absolutely essential and in the process impartial nature of the evaluator is required in the examination system to become popular amongst the students. It is a wrong approach or concept to award the students by way of easy marking to get cheap popularity among the students to which they do not deserve. It is a primary responsibility of the teacher that right students who are really putting up lot of hard work with right kind of intelligence are correctly awarded.

The marking patterns should be justifiable to the students without any ambiguity and teacher should see that students are faced with unjust circumstances.