JAVA PROGRAMMING

RECORD

B.TECH (R-18 Regulation)
(II YEAR – II SEM)
(2020-21)

DEPARTMENT OF
COMPUTER SCIENCE AND ENGINEERING

MALLA REDDY COLLEGE OF ENGINEERING & TECHNOLOGY
(Autonomous Institution – UGC, Govt. of India)
Recognized under 2(f) and 12 (B) of UGC ACT 1956
(Affiliated to JNTUH, Hyderabad, Approved by AICTE - Accredited by NBA & NAAC – 'A' Grade - ISO 9001:2015 Certified)
Maisammaguda, Dhulapally (Post Via. Hakimpet), Secunderabad – 500100, Telangana State, India

Department of Computer Science and Engineering

Vision

- To acknowledge quality education and instill high patterns of discipline making the students technologically superior and ethically strong which involves the improvement in the quality of life in human race.

Mission

- To achieve and impart holistic technical education using the best of infrastructure, outstanding technical and teaching expertise to establish the students in to competent and confident engineers.
- Evolving the center of excellence through creative and innovative teaching learning practices for promoting academic achievement to produce internationally accepted competitive and world class professionals.

PROGRAMME EDUCATIONAL OBJECTIVES (PEOs)

PEO1 – ANALYTICAL SKILLS

1. To facilitate the graduates with the ability to visualize, gather information, articulate, analyze, solve complex problems, and make decisions. These are essential to address the challenges of complex and computation intensive problems increasing their productivity.

PEO2 – TECHNICAL SKILLS

2. To facilitate the graduates with the technical skills that prepare them for immediate employment and pursue certification providing a deeper understanding of the technology in advanced areas of computer science and related fields, thus encouraging to pursue higher education and research based on their interest.

PEO3 – SOFT SKILLS

3. To facilitate the graduates with the soft skills that include fulfilling the mission, setting goals, showing self-confidence by communicating effectively, having a positive attitude, get involved in team-work, being a leader, managing their career and their life.

PEO4 – PROFESSIONAL ETHICS

To facilitate the graduates with the knowledge of professional and ethical responsibilities by paying attention to grooming, being conservative with style, following dress codes, safety codes,and adapting themselves to technological advancements.

PROGRAM SPECIFIC OUTCOMES (PSOs)

After the completion of the course, B. Tech Computer Science and Engineering, the graduates will have the following Program Specific Outcomes:

1. **Fundamentals and critical knowledge of the Computer System:-** Able toUnderstand the working principles of the computer System and its components , Apply the knowledge to build, asses, and analyze the software and hardware aspects of it .

2. **The comprehensive and Applicative knowledge of Software Development:** Comprehensive skills of Programming Languages, Software process models, methodologies, and able to plan, develop, test, analyze, and manage the software and hardware intensive systems in heterogeneous platforms individually or working in teams.

3. **Applications of Computing Domain & Research:** Able to use the professional,managerial, interdisciplinary skill set, and domain specific tools in development processes, identify the research gaps, and provide innovative solutions to them.

PROGRAM OUTCOMES (POs)

Engineering Graduates will be able to:

1. **Engineering knowledge**: Apply the knowledge of mathematics, science, engineeringfundamentals, and an engineering specialization to the solution of complex engineering problems.
2. **Problem analysis**: Identify, formulate, review research literature, and analyzecomplex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.
3. **Design / development of solutions**: Design solutions for complex engineeringproblems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.
4. **Conduct investigations of complex problems**: Use research-based knowledge andresearch methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
5. **Modern tool usage**: Create, select, and apply appropriate techniques, resources, andmodern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.
6. **The engineer and society**: Apply reasoning informed by the contextual knowledge toassess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.
7. **Environment and sustainability**: Understand the impact of the professionalengineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.
8. **Ethics**: Apply ethical principles and commit to professional ethics and responsibilitiesand norms of the engineering practice.
9. **Individual and team work**: Function effectively as an individual, and as a member orleader in diverse teams, and in multidisciplinary settings.
10. **Communication**: Communicate effectively on complex engineering activities withthe engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.
11. **Project management and finance**: Demonstrate knowledge and understanding ofthe engineering and management principles and apply these to one’s own work, as a member and leader in a team, to manage projects and in multi disciplinary environments.
12. **Life- long learning**: Recognize the need for, and have the preparation and ability toengage in independent and life-long learning in the broadest context of technological change.

1. Lab Objectives:

- To prepare students to become familiar with the Standard Java technologies of J2SE
- To prepare students to excel in Object Oriented programming and to succeed as a Java Developer through global rigorous education.
- To provide Students with a solid foundation in OOP fundamentals required to solve programming problems and also to learn Advanced Java topics like J2ME, J2EE, JSP, JavaScript
- To train Students with good OOP programming breadth so as to comprehend, analyze, design and create novel products and solutions for the real life problems.
- To inculcate in students professional and ethical attitude, multidisciplinary approach and an ability to relate java programming issues to broader application context.
- To provide student with an academic environment aware of excellence, written ethical codes and guidelines and lifelong learning needed for a successful professional career.

2. Lab Outcomes:

Upon successful completion of this course, the students will be able to:

- Able to analyze the necessity for Object Oriented Programming paradigm and over structured programming and become familiar with the fundamental concepts in OOP.
- Demonstrate an ability to design and develop java programs, analyze, and interpret object oriented data and report results.
- Demonstrate an ability to design an object oriented system, AWT components or multithreaded process as per needs and specifications.
- Demonstrate an ability to visualize and work on laboratory and multidisciplinary tasks like console and windows applications both for standalone and Applets programs

3. Introduction about lab

There are 65 systems installed in this Lab. Their configurations are as follows:

- **Hardware / Software's installed: I**ntel® CORE™ i3-3240 CPU@3.40GHZ RAM:4GB / C, C++ Compiler ,JAVA JDK 1.8,EditPlus.
- Systems are provided for students in the **1:1 ratio.**
- Systems are assigned numbers and same system is allotted for students when they do the lab.
- All systems are configured in **DUAL BOOT** mode i.e., Students can boot from Windows XP or Linux as per their lab requirement. This is very useful for
    - students because they are familiar with different Operating Systems so that they can execute their programs in different programming environments.

4. Guidelines to students

A. Standard operating procedure

a) Explanation on today's experiment by the concerned faculty using PPT covering the following aspects:

1) Name of the experiment
2) Aim
3) Software/Hardware requirements

b) Writing the java programs by the students

c) Commands for executing programs 100 mins.

Writing of the experiment in the Observation Book

The students will write the today's experiment in the Observation book as per the following format:

a) Name of the experiment
b) Aim
c) Software/Hardware required
d) Writing the program
e) Viva-Voce Questions and Answers
f) Errors observed (if any) during compilation/execution
g) Signature of the Faculty

h) Viva-Voce Questions and Answers
i) Errors observed (if any) during compilation/execution
j) Signature of the Faculty

B. Guide Lines to Students in Lab

Disciplinary to be maintained by the students in the Lab

- Students are required to carry their lab observation book and record book with completed experiments while entering the lab.
- Students must use the equipment with care. Any damage is caused student is punishable
- Students are not allowed to use their cell phones/pen drives/ CDs in labs.
- Students need to be maintain proper dress code along with ID Card
- Students are supposed to occupy the computers allotted to them and are not supposed to talk or make noise in the lab.
- Students, after completion of each experiment they need to be updated in observation notes and same to be updated in the record.
- Lab records need to be submitted after completion of experiment and get it corrected with the concerned lab faculty.
- If a student is absent for any lab, they need to be completed the same experiment in the free time before attending next lab.

Steps to perform experiments in the lab by the student

Step1: Students have to write the date, aim, Software & Hardware requirements for that experiment in the observation book.

Step2: Students have to listen and understand the experiment explained by the faculty and note down the important points in the observation book. Step3: Students need to write procedure/algorithm in the observation book. Step4: Analyze and Develop/implement the logic of the program by the student in respective platform Step5: after approval of logic of the experiment by the faculty then the experiment has to be executed on the system.

Step6: After successful execution the results are to be shown to the faculty and noted the same in the observation book.

Step7: Students need to attend the Viva-Voce on that experiment and write the same in the observation book.

Step8: Update the completed experiment in the record and submit to the concerned faculty in-charge.

Instructions to maintain the record

- Before start of the first lab they have to buy the record and bring the record to the lab.
- Regularly (Weekly) update the record after completion of the experiment and get it corrected with concerned lab in-charge for continuous evaluation.
- In case the record is lost inform the same day to the faculty in charge and get the new record within 2 days the record has to be submitted and get it corrected by the faculty.
- If record is not submitted in time or record is not written properly, the evaluation marks (5M) will be deducted.

Awarding the marks for day to day evaluation

Total marks for day to day evaluation is 15 Marks as per Autonomous (JNTUH).

These 15 Marks are distributed as:

Record	5 Marks
Exp setup/program written	5 Marks
Result and Viva-Voce	5 Marks

Allocation of Marks for Lab Internal

Total marks for lab internal are 30 Marks as per Autonomous (JNTUH.)

These 30 Marks are distributed as:

Average of day to day evaluation marks: 20 Marks

Lab Mid exam: 10 Marks

Allocation of Marks for Lab External

Total marks for lab Internal and External are 70 Marks as per Autonomous / (JNTUH.)

These 70 External Lab Marks are distributed as:

Program Written	30 Marks
Program Execution and Result	20 Marks
Viva-Voce	10 Marks
Record	10 Marks

C. General laboratory instructions

1. Students are advised to come to the laboratory at least 5 minutes before (to the starting time), those who come after 5 minutes will not be allowed into the lab.
2. Plan your task properly much before to the commencement, come prepared to the lab with the synopsis / program / experiment details.
3. Student should enter into the laboratory with:
    a. Laboratory observation notes with all the details (Problem statement, Aim, Algorithm, Procedure, Program, Expected Output, etc.,) filled in for the lab session.
    b. Laboratory Record updated up to the last session experiments and other utensils (if any) needed in the lab.
    c. Proper Dress code and Identity card.

4. Sign in the laboratory login register, write the TIME-IN, and occupy the computer system allotted to you by the faculty.
5. Execute your task in the laboratory, and record the results / output in the lab observation note book, and get certified by the concerned faculty.
6. All the students should be polite and cooperative with the laboratory staff, must maintain the discipline and decency in the laboratory.
7. Computer labs are established with sophisticated and high end branded systems, which should be utilized properly.
8. Students / Faculty must keep their mobile phones in SWITCHED OFF mode during the lab sessions. Misuse of the equipment, misbehaviors with the staff and systems etc., will attract severe punishment.
9. Students must take the permission of the faculty in case of any urgency to go out ; if anybody found loitering outside the lab / class without permission during working hours will be treated seriously and punished appropriately.
10. Students should LOG OFF/ SHUT DOWN the computer system before he/she leaves the lab after completing the task (experiment) in all aspects. He/she must ensure the system / seat is kept properly.

Head of the Department **Principal**

INDEX

S.No	List of Programs	Page Nos.	Date	Signature
1	Write a java program to find the Fibonacci series using recursive and non recursive functions			
2	Write a java program to multiply two given matrices.			
3	Write a java program for Method overloading and Constructor overloading			
4	write a java program that prompts the user for an integer and then printouts all prime numbers up to that integer			
5	Write a java program to display the employee details using Scanner class			
6	Write a java program that checks whether a given string is palindrome or not			
7	A)Write a java program to represent Abstract class with example.			
	B)Write a java program to implement Interface using extends keyword			
8	A)Write a java program to create user defined package			
	B)Write a java program to create inner classes			
9	A)Write a java program for creating multiple catch blocks			
	B)Write a java program for producer and consumer problem using Threads			
10	Write a Java program that implements a multi-thread application that has three threads			
11	A)Write a java program to display File class properties			
	B)Write a java program to represent ArrayList class			
	C)Write a Java program loads phone no, name from a text file using hash table			
12	Write an applet program that displays a simple message			
13	A)Write a Java program compute factorial value using Applet			
	B)Write a program for passing parameters using Applet			
14	Write a java program for handling Mouse events and Key events			
15	Write a java program that connects to a database using JDBC			
16	A)Write a java program to connect to a database using JDBC and insert values into it			
	B)Write a java program to connect to a database using JDBC and delete values from it			
17	Write a java program that works as a simple calculator. Use a Grid Layout to arrange Buttons for digits and for the + - * % operations. Add a text field to display the result			

PROGRAM-1 Date:

Aim: Write a java program to find the Fibonacci series using recursive and non recursive functions

Program:

```
//Class to write the recursive and non recursive functions.
class fib
{
int a,b,c;
//Non recursive function to find the Fibonacciseries.
void nonrecursive(int n)
{
a=0;
b=1;
c=a+b;
System.out.print(b);
while(c<=n)
{
System.out.print(c);
a=b;
b=c;
c=a+b;
}
}
// Recursive function to find the Fibonacci series.
int recursive(int n)
{
if(n==0)
            return (0);
             if(n==1)
                        return (1);
            else
                        return(recursive(n-1)+recursive(n-2));
}
}
// Class that calls recursive and non recursive functions
 . class fib1
{
public static void main(String args[])
{
int n;
// Accepting the value of n at run
 time. n=Integer.parseInt(args[0]);
System.out.println("the recursion using non recursive is"); // Creating object for the fib
class.fib f=new fib();
// Calling non recursive function of
 fib class. f.nonrecursive(n);
System.out.println("the recursion using recursive is"); ffor(int i=0;i<=n;i++)
{
// Calling recursive function of fib class. int F1=f.recursive(i);
System.out.print(F1);
}}}
```

Three Test Outputs:

EXERCISE:

1. Write a java program to print the multiplication table .
2. Write a java program to find the Factorial of a given integer using recursive and non recursive functions

RECORD NOTES

PROGRAM -2 Date:

Aim: Write a java program to multiply two given matrices.

```
// Class to find multiplication of matrices.
class matri
{
public static void main(String args[])
{
// Accept the number of rows and columns at run time.
int m=Integer.parseInt(args[0]);
int n=Integer.parseInt(args[1]);
// Initialize the arrays.
int a[][]=new int[m][n]; int b[][]=new int[m][n]; int c[][]=new int[m][n]; int i=2;
// Loop to accept the values into a
matrix. for(int j=0;j<m;j++)
{for(int k=0;k<n;k++)
{
a[j][k]=Integer.parseInt(args[i]);
i++;
}
}
// Loop to accept the values into b matrix.
for(int j=0;j<m;j++)
{
for(int k=0;k<n;k++)
{
        b[j][k]=Integer.parseInt(args[i]);
        i++;
}
}
// Loop to multiply two matrices .
for(int j=0;j<m;j++)
{
for(int k=0;k<n;k++)
  {
c[j][k]=0;
for(int l=0;l<m;l++)
{
        c[j][k]=c[j][k]+(a[j][l]*b[l][k]);
}
}
}
// Loop to display the result .
for(int j=0;j<m;j++)
{
for(int k=0;k<n;k++)
{
System.out.print(c[j][k]);
}
System.out.println();
}
}
}
```

Three test outputs:

RECORD NOTES

PROGRAM -3 Date:

Aim: Write a java program that prompts the user for an integer and then printouts all prime numbers up to that integer

```
import java.lang.*;
class Prime
{
public static void main(String arg[])
{
int n,c,i,j;
n=Integer.parseInt(arg[0]);
System.out.println("prime numbers are");
for(i=1;i<=n;i++)
{
c=0;
for(j=1;j<=i;j++)
{
if(i%j==0)
c++;
}
if(c==2)
System.out.println(" "+i);
}
}
}
```

Three test outputs:

EXERCISE:

1. Write a java program to find all even and odd integers up to a given integer.
2. Write a java program to add and subtract two given matrices.
3. Write a java program that reads a line of integers and displays each integers and the product of all integers use String Tokenizer.

RECORD NOTES

PROGRAM -4 **Date:**

Aim: Write a java program that checks whether a given string is palindrome or not Program:

```
// Class to find whether string is palindrome or not.
class palindrome
{
public static void main(String args[])
{
// Accepting the string at run time.
 String s=args[0];
String s1=""; int l,j;
// Finding the length of the string.
l=s.length();
// Loop to find the reverse of the
string. for(j=l-1;j>=0;j--)
{
s1=s1+s.charAt(j);
}
// Condition to find whether two strings are equal // and display the
message. if(s.equals(s1))
System.out.println("String "+s+" is palindrome");
else
System.out.println("String "+s+" is not palindrome");
}

}
```

Three test outputs:

EXERCISE:

1. Write a java program to sort the given integers in ascending/descending order.
2. Write a java program to display characters in a string in sorted order.
3. Write a program that uses a sequence inputstream to output the contents of two files.
4. Write a java program that reads a file and displays the file on the screen, with an asterisk mark before each line.
5. Write a java program that displays the number of characters, lines, words, white spaces in a text file.

RECORD NOTES

PROGRAM -5 Date:

Write a java program to display the employee details using Scanner class

```
importjava.util.*;
classEmployeeDetails
{
public static void main(String args[])
{
System.out.println("enter name,id,age,salary");
Scanner sc=new Scanner(System.in);
String n=sc.next();
int i=sc.nextInt();
int a=sc.nextInt();
float s=sc.nextFloat();
System.out.println("name is"+n+"idis"+i+"ageis"+a+"salaryis"+s);
}
}
```

Three test Outputs:

EXERCISE: Write a java program to Read and display the student details using Scanner class.

RECORD NOTES

PROGRAM -6 Date:

Write a java program that checks whether a given string is palindrome or not

```
class Palindrome
{
public static void main(String args[])
{
// Accepting the string at run time.
String s=args[0];
String s1="";
intl,j;
// Finding the length of the string.
l=s.length();
// Loop to find the reverse of the string.
for(j=l-1;j>=0;j--)
{
s1=s1+s.charAt(j);
}
// Condition to find whether two strings are equal
if(s.equals(s1))
System.out.println("String "+s+" is palindrome");
else
System.out.println("String "+s+" is not palindrome");
}
}
```

Three Test Outputs:

RECORD NOTES

PROGRAM -7 A Date:

Write a java program to represent Abstract class with example

```
abstract class Shape
{
abstract void numberOfSides();
}
// Classes that illustrates the abstract method.
class Trapezoid
{
voidnumberOfSides()
{
System.out.println("The no. of side's in trapezoidal are6");
}
}
class Triangle
{
voidnumberOfSides()
{
System.out.println("The no. of side's in triangle are:3 ");
}
}
classHexogon
{
voidnumberOfSides()
{ System.out.println("The no. of side's in hexogon are:6 ");
}
}
// Class that create objects and call the method.
classShapeDemo
{
public static void main(String args[])
{
Trapezoid obj1 = new Trapezoid();
Triangle obj2 = new Triangle();
Hexogon obj3 = new Hexogon();
obj1.numberOfSides();
obj2.numberOfSides();
obj3.numberOfSides(); }
}
```

Three Test Outputs:

RECORD NOTES

PROGRAM -7B   Date:

Write a java program to implement Interface using extends keyword

```
class Person
  {
    String name;
Person(String n)
    {
name = "Person: " + n;
    }
 }
interface Mother
  {
public void FeedChildren();
  }
interface Wife
  {
public void CallHusband();
  }
classWifeAndMother extends Person implements Wife, Mother
  {
WifeAndMother(String n)
     {
super(n);
name = "Wife and mother: " + n;
     }
public void FeedChildren()
  {
System.out.println(name + " is feeding the children.");
   }
public void CallHusband()
  {
System.out.println(name + " is calling her husband.");
  }
}
public class Test5
 {
public static void main(String args[])
      {
          Person p = new Person("Kiran");
WifeAndMother w = new WifeAndMother("Radha");
System.out.println("p is a " + p.name);
System.out.println("w is a " + w.name);
w.FeedChildren();
w.CallHusband();
      }
```

Three Test Outputs:

RECORD NOTES

PROGRAM -8 A Date:

Write a java program to create user defined package

```
package pack;
public class A{
public void msg(){System.out.println("Hello");}
}

import pack.*;
class B{
public static void main(String args[]){
  A obj = new A();
obj.msg();
 }
}
```

Three Test Outputs:

RECORD NOTES

PROGRAM -8B Date:

Write a java program to create inner classes

```
class A
{
int a=10;
void display()
{
B b=new B();
b.show();
}
class B
{
int b=20;
void show()
{
System.out.println(" a value is " +a);
System.out.println(" b value is " +b);
}
}
}
classInnerDemo
{
public static void main(String args[])
{
A a=new A();
a.display();
}
}
```

Three Test Outputs:

RECORD NOTES

PROGRAM -9A Date:

Write a java program for creating multiple catch blocks

```
public class MultipleCatchBlocks {

public static void main(String[] args) {

try{
int a[]=new int[5];
a[5]=30/0;
            }
catch(ArithmeticException e)
              {
System.out.println("Arithmetic Exception occurs");
              }
catch(ArrayIndexOutOfBoundsException e)
              {
System.out.println("ArrayIndexOutOfBounds Exception occurs");
              }
catch(Exception e)
              {
System.out.println("Parent Exception occurs");
              }
System.out.println("rest of the code");
   }
}
```

Three Test Outputs:

RECORD NOTES

PROGRAM -9B   Date:

Write a java program for producer and consumer problem using Threads

```
classInterThreadDemo
{
public static void main(String args[])
{
Producer p1=new Producer();
Consumer c1=new Consumer(p1);
Thread t1=new Thread(p1);
Thread t2=new Thread(c1);
t2.start();
t1.start();
}
}
class Producer extends Thread
{
StringBuffersb;
Producer()
{
sb=new StringBuffer();
}
public void run()
{
synchronized(sb)
{
for(int i=0;i<=10;i++)
{
try
{
sb.append(i+":");
Thread.sleep(1000);
System.out.println("appending");
}
catch(InterruptedException e)
{
System.out.println(e);
}
}
sb.notify();
}
}
}

class Consumer extends Thread
{
Producer prod;
Consumer(Producer prod)
{
this.prod=prod;
}

public void run()
{
synchronized(prod.sb)
{
try
{
prod.sb.wait();
}
catch(Exception e)
{
System.out.println(e);
}
System.out.println(prod.sb);
}
}
}
```

Three Test Outputs:

RECORD NOTES

PROGRAM -10 Date:

Write a Java program that implements a multi-thread application that has three threads

```
class Thread1 extends Thread
{
public void run()
{
for(int i=0;i<=5;i++)
{
System.out.println("Thread1:"+ i);
}
}
}
class Thread2 extends Thread
{
public void run()
{
for(int j=0;j<=5;j++)
{
System.out.println("Thread2:"+ j);
}
}
}
class Thread3 extends Thread
{
public void run()
{
for(int k=0;k<=5;k++)
{
System.out.println("Thread3:"+ k);
}
}
}
classMultiThreadDemo
{
public static void main(String args[])
{
Thread1 t1=new Thread1();
Thread2 t2=new Thread2();
Thread3 t3=new Thread3();
t1.start();
t2.start();
t3.start();
for(int i=0;i<=5;i++)
{
System.out.println("main thread:"+ i);
}
}
}
```

Three Test Outputs:

RECORD NOTES

PROGRAM -11A Date:

Write a java program to display File class properties

```
import java.io.*;
public class FileDemo2 {
public static void main(String[] args) {
        String fname=args[0];
File f=new File(fname);
System.out.println("path"+f.getPath());
System.out.println("parent"+f.getAbsolutePath());
System.out.println("parent"+f.getParent());
System.out.println("exits"+f.exists());
if(f.exists())
{
System.out.println("isWritable"+f.canWrite());
System.out.println("isReadable"+f.canRead());
System.out.println("isDirectory"+f.isDirectory());
System.out.println("size of the file"+f.length());
}
}
}
```

Three Test Outputs:

RECORD NOTES

PROGRAM -11B Date:

Write a java program to represent ArrayList class

```
importjava.util.*;
classTestJavaCollection{
public static void main(String args[]){
ArrayList<String> list=new ArrayList<String>();//Creating arraylist
list.add("Ravi");//Adding object in arraylist
list.add("Vijay");
list.add("Ravi");
list.add("Ajay");
//Traversing list through Iterator
Iterator itr=list.iterator();
while(itr.hasNext()){
System.out.println(itr.next());
}
}
}
```

Three test outputs

RECORD NOTES

PROGRAM -11C Date:

Write a Java program loads phone no, name from a text file using hash table

```
import java.util.*;
class HTDemo {
public static void main(String args[]) {
Hashtable balance = new Hashtable();
Enumeration names;
String str;
doublebal;
balance.put("John Doe", new Double(3434.34));
balance.put("Tom Smith", new Double(123.22));
balance.put("Jane Baker", new Double(1378.00));
balance.put("Todd Hall", new Double(99.22));
balance.put("Ralph Smith", new Double(19.08));
// Show all balances in hash table.
names = balance.keys();
while(names.hasMoreElements()) {
str = (String) names.nextElement();
System.out.println(str + ": " +
balance.get(str));
}
System.out.println();
// Deposit 1,000 into John Doe's account
bal = ((Double)balance.get("John Doe")).doubleValue();
balance.put("John Doe", new Double(bal+1000));
System.out.println("John Doe's new balance: " +
balance.get("John Doe"));
}
```

Three test outputs

RECORD NOTES

PROGRAM -12 Date:

Write an applet program that displays a simple message

```
importjava.awt.*;
importjava.applet.*;
/*
<applet code="FirstApplet" width=200 height=300>
</applet>*/
public class FirstApplet extends Applet
{
public void init()
{
setBackground(Color.red);
}
public void paint(Graphics g)
{
g.drawString("this is first applet",50,30);
showStatus("welcome");
}
}
```

Three test outputs:

RECORD NOTES

PROGRAM -13A Date:

Write a Java program compute factorial value using Applet

```
importjava.awt.*;
importjava.awt.event.*;
importjava.applet.*;
/*
<applet code="FactorialApplet" width=300 height=300>
</applet>
*/
public class FactorialApplet extends Applet implements ActionListener
{
Label L1,L2;
TextField T1,T2;
Button B1;
public void init()
{
setLayout(new FlowLayout(FlowLayout.LEFT));
L1=new Label("enter the value");
add(L1);
T1=new TextField(10);
add(T1);
L2=new Label("factorial value is");
add(L2);
T2=new TextField(10);
add(T2);
B1=new Button("compute");
add(B1);
B1.addActionListener(this);
}
public void actionPerformed(ActionEvent e)
{
if(e.getSource()==B1)
{
int value=Integer.parseInt(T1.getText());
int fact=factorial(value);
T2.setText(String.valueOf(fact));
}
}
int factorial(int n)
{
if(n==0)
return 1;
else
return n*factorial(n-1);
}
}
```

Three test outputs:

RECORD NOTES

PROGRAM -13B Date:

Write a program for passing parameters using Applet

```
importjava.awt.*;
importjava.applet.*;
/*
<applet code="MyApplet" width=200 height=300>
<param name="t1" value="Ravi"><param name="t2" value="102">
</applet>
*/
public class MyApplet extends Applet
{
String n;
String id;
public void init()
{
n=getParameter("t1");
id=getParameter("t2");
}
public void paint(Graphics g)
{
g.drawString("name is"+n,100,100);
g.drawString("id is"+id,100,150);
}
}
```

Three test outputs:

RECORD NOTES

PROGRAM -14 Date:

Write a java program for handling Mouse events

```
importjava.awt.*;
importjava.awt.event.*;
importjava.applet.*;
/*
<applet code="MouseEvents" width=300 height=100>
</applet>
*/
public class MouseEvents extends Applet implements MouseListener,
MouseMotionListener {
String msg = "";
intmouseX = 0, mouseY = 0; // coordinates of mouse
public void init() {
addMouseListener(this);
addMouseMotionListener(this);
}
// Handle mouse clicked.
public void mouseClicked(MouseEvent me) {
// save coordinates
mouseX = 0;
mouseY = 10;
msg = "Mouse clicked.";
repaint();
}
// Handle mouse entered.
public void mouseEntered(MouseEvent me) {
// save coordinates
mouseX = 0;
mouseY = 10;
msg = "Mouse entered.";
repaint();
}
// Handle mouse exited.
public void mouseExited(MouseEvent me) {
// save coordinates
mouseX = 0;
mouseY = 10;
msg = "Mouse exited.";
repaint();
}
// Handle button pressed.
public void mousePressed(MouseEvent me) {
// save coordinates
mouseX = me.getX();
mouseY = me.getY();
msg = "Down";
repaint();
}
// Handle button released.
public void mouseReleased(MouseEvent me) {
// save coordinates
mouseX = me.getX();
mouseY = me.getY();
msg = "Up";
repaint();
}
// Handle mouse dragged.
public void mouseDragged(MouseEvent me) {
// save coordinates
mouseX = me.getX();
mouseY = me.getY();
msg = "*";
showStatus("Dragging mouse at " + mouseX + ", " + mouseY);
repaint();
}
// Handle mouse moved.
public void mouseMoved(MouseEvent me) {
// show status
showStatus("Moving mouse at " + me.getX() + ", " + me.getY());
}
// Display msg in applet window at current X,Y location.
public void paint(Graphics g) {
g.drawString(msg, mouseX, mouseY);
}
}
```

Three test outputs:

RECORD NOTES

Write a program for handling Key Events

```
importjava.awt.*;
importjava.awt.event.*;
importjava.applet.*;
/*
<applet code="SimpleKey" width=300 height=100>
</applet> */
public class SimpleKey extends Applet implements KeyListener {
String msg = "";
int X = 10, Y = 20; // output coordinates
public void init() {
addKeyListener(this);
requestFocus(); // request input focus
}
public void keyPressed(KeyEventke) {
showStatus("Key Down");
}
public void keyReleased(KeyEventke) {
showStatus("Key Up");
}
public void keyTyped(KeyEventke) {
msg += ke.getKeyChar();
repaint();
}
    // Display keystrokes.
public void paint(Graphics g) {
g.drawString(msg, X, Y);
}
}
```

Three test outputs

RECORD NOTES

PROGRAM -15 Date:

Aim: Write a java program that connects to a database using JDBC

Program:

```
import java.sql.Connection;
import java.sql.DriverManager;
public class PostgreSQLJDBC
 {
   public static void main(String args[])
{
      Connection c = null;
      try {
         Class.forName("org.postgresql.Driver");
         c = DriverManager .getConnection("jdbc:postgresql://localhost:5432/testdb",
            "postgres", "123");
      } catch (Exception e) {
         e.printStackTrace();
         System.err.println(e.getClass().getName()+": "+e.getMessage());
         System.exit(0);
      }
      System.out.println("Opened database successfully");
   }
}
```

Three test outputs:

RECORD NOTES

PROGRAM -16A Date:

Write a java program to connect to a database using JDBC and insert values into it

```
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.Statement;
public class PostgreSQLJDBC
{
  public static void main(String args[])
{
    Connection c = null;
    Statement stmt = null;
    try {
      Class.forName("org.postgresql.Driver");
      c = DriverManager
        .getConnection("jdbc:postgresql://localhost:5432/testdb",
        "manisha", "123");
      c.setAutoCommit(false);
      System.out.println("Opened database successfully");
      stmt = c.createStatement();
      String sql = "INSERT INTO COMPANY (ID,NAME,AGE,ADDRESS,SALARY) "
          + "VALUES (1, 'Paul', 32, 'California', 20000.00 );";
     stmt.executeUpdate(sql);

      sql = "INSERT INTO COMPANY (ID,NAME,AGE,ADDRESS,SALARY) "
          + "VALUES (2, 'Allen', 25, 'Texas', 15000.00
      );"; stmt.executeUpdate(sql);

      sql = "INSERT INTO COMPANY (ID,NAME,AGE,ADDRESS,SALARY) "
          + "VALUES (3, 'Teddy', 23, 'Norway', 20000.00
      );"; stmt.executeUpdate(sql);

      sql = "INSERT INTO COMPANY (ID,NAME,AGE,ADDRESS,SALARY) "
          + "VALUES (4, 'Mark', 25, 'Rich-Mond ', 65000.00
      );"; stmt.executeUpdate(sql);

      stmt.close();
      c.commit();
      c.close();
    } catch (Exception e) {
      System.err.println( e.getClass().getName()+": "+ e.getMessage()
      ); System.exit(0);
    }
    System.out.println("Records created successfully");
  }
}
```

Three test outputs:

RECORD NOTES

PROGRAM -16B Date:

Write a java program to connect to a database using JDBC and delete values from it import

```
java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.Statement;
public class PostgreSQLJDBC6 {
  public static void main( String args[] )
    {
     Connection c = null;
     Statement stmt = null;
     try {
     Class.forName("org.postgresql.Driver");
       c = DriverManager
          .getConnection("jdbc:postgresql://localhost:5432/testdb",
          "manisha", "123");
       c.setAutoCommit(false);
       System.out.println("Opened database successfully");

       stmt = c.createStatement();
       String sql = "DELETE from COMPANY where
       ID=2;"; stmt.executeUpdate(sql);
       c.commit();
       ResultSet rs = stmt.executeQuery( "SELECT * FROM COMPANY;"
       ); while ( rs.next() ) {

          int id = rs.getInt("id");
          String name = rs.getString("name");
          int age = rs.getInt("age");
          String address = rs.getString("address");
          float salary = rs.getFloat("salary");
          System.out.println( "ID = " + id );
          System.out.println( "NAME = " + name );
          System.out.println( "AGE = " + age );
          System.out.println( "ADDRESS = " + address );
          System.out.println( "SALARY = " + salary );
          System.out.println();
       }
       rs.close();
       stmt.close();
       c.close();
     } catch ( Exception e ) {
       System.err.println( e.getClass().getName()+": "+ e.getMessage()
       ); System.exit(0);
     }
     System.out.println("Operation done successfully");
    }
}
```

Three test outputs:

RECORD NOTES

PROGRAM -17 Date:

Write a java program that works as a simple calculator. Use a Grid Layout to arrange Buttons for digits and for the + - * % operations. Add a text field to

```
importjava.awt.*;
importjava.awt.event.*;
importjava.applet.*;
importjavax.swing.*;
/*
<applet code="Calculator" width=300 height=300>
</applet>
*/
public class Calculator extends JAppletimplements ActionListener
{
        String msg=" ";
        int v1,v2,result;
        JTextField t1;
        JButtonb[]=new JButton[10];
        JButtonadd,sub,mul,div,clear,mod,EQ;
        char OP;
        public void init()
        {
Container contentPane = getContentPane();
contentPane.setLayout(new FlowLayout());
                Color k=new Color(120,89,90);
                setBackground(k);
                t1=new JTextField(10);
                GridLayoutgl=new GridLayout(4,5);
                setLayout(gl);
                for(int i=0;i<10;i++)
                {
                        b[i]=new JButton(""+i);
                }
                add=new JButton("add");
                sub=new JButton("sub");
                mul=new JButton("mul");
                div=new JButton("div");
                mod=new JButton("mod");
                clear=new JButton("clear");
                EQ=new JButton("EQ");
                t1.addActionListener(this);
                add(t1);
                for(int i=0;i<10;i++)
                {
                        add(b[i]);
                }
                contentPane.add(add);
                contentPane.add(sub);
                contentPane.add(mul);
                contentPane.add(div);
        contentPane.add(
        mod);
        contentPane.add(c
        lear);
        contentPane.add(E
        Q); for(int
        i=0;i<10;i++)
        {
                b[i].addActionListener(this);
        }
        add.addActionListener(t
        his);
        sub.addActionListener(t
        his);
        mul.addActionListener(t
        his);
        div.addActionListener(t
        his);
        mod.addActionListener(
        this);
        clear.addActionListener(
        this);
        EQ.addActionListener(t
        his);
}

public void actionPerformed(ActionEventae)
{
        String
        str=ae.getActionCommand();
        charch=str.charAt(0);
        if (
        Character.isDigit(ch))
        t1.setText(t1.getText()
        +str); else
        if(str.equals("add"))
        {
                v1=Integer.parseInt(t1.getTe
                xt()); OP='+';
                t1.setText("");
        }
        else if(str.equals("sub"))
        {
                v1=Integer.parseInt(t1.getTe
                xt()); OP='-';
                t1.setText("");
```

```
                    }
                    else if(str.equals("mul"))
                    {
                              v1=Integer.parseInt(t1.getTe
                              xt()); OP='*';
                              t1.setText("");
                    }
                    else if(str.equals("div"))
                    {
                              v1=Integer.parseInt(t1.getTe
                              xt()); OP='/';
                              t1.setText("");
                    }
                    else if(str.equals("mod"))
                    {
                              v1=Integer.parseInt(t1.getTe
                              xt()); OP='%';
                              t1.setText("");
                    }
                    if(str.equals("EQ"))
                    {
                              v2=Integer.parseInt(t1.getTe
                              xt()); if(OP=='+')
                                        result=v
                              1+v2; else
                              if(OP=='-')
                                        result=v
                              1-v2; else
                              if(OP=='*')
                                        result=v
                              1*v2; else
                              if(OP=='/')
                                        result=v
                              1/v2; else
                              if(OP=='%')
                                        result=v1
                              %v2;
                              t1.setText(""+res
                              ult);
                    }
                    if(str.equals("clear"))
                    {
                              t1.setText("");
                    }
          }
}
```

Three test outputs:

RECORD NOTES