**EXPERIMENT No. 1**

**History:**

Java was started as a project called "Oak" by James Gosling in June 1991. Gosling's goals were to implement a virtual machine and a language that had a familiar C-like notation but with greater uniformity and simplicity than C/C++. The first public implementation was Java 1.0 in 1995. It made the promise of "Write Once, Run Anywhere", with free runtimes on popular platforms. It was fairly secure and its security was configurable, allowing for network and file access to be limited. The major web browsers soon incorporated it into their standard configurations in a secure "applet" configuration popular quickly. New versions for large and small platforms (J2EE and J2ME) soon were designed with the advent of "Java 2". Sun has not announced any plans for a "Java 3".story:-

**Features:**

Following are the notable features of Java:

**Object Oriented:**

In Java, everything is an Object. Java can be easily extended since it is based on the Object model.

**Platform Independent:**

Unlike many other programming languages including C and C++, when Java is compiled, it is not compiled into platform specific machine, rather into platform-independent byte code. This byte code is distributed over the web and interpreted by the Virtual Machine (JVM) on whichever platform it is being run on.

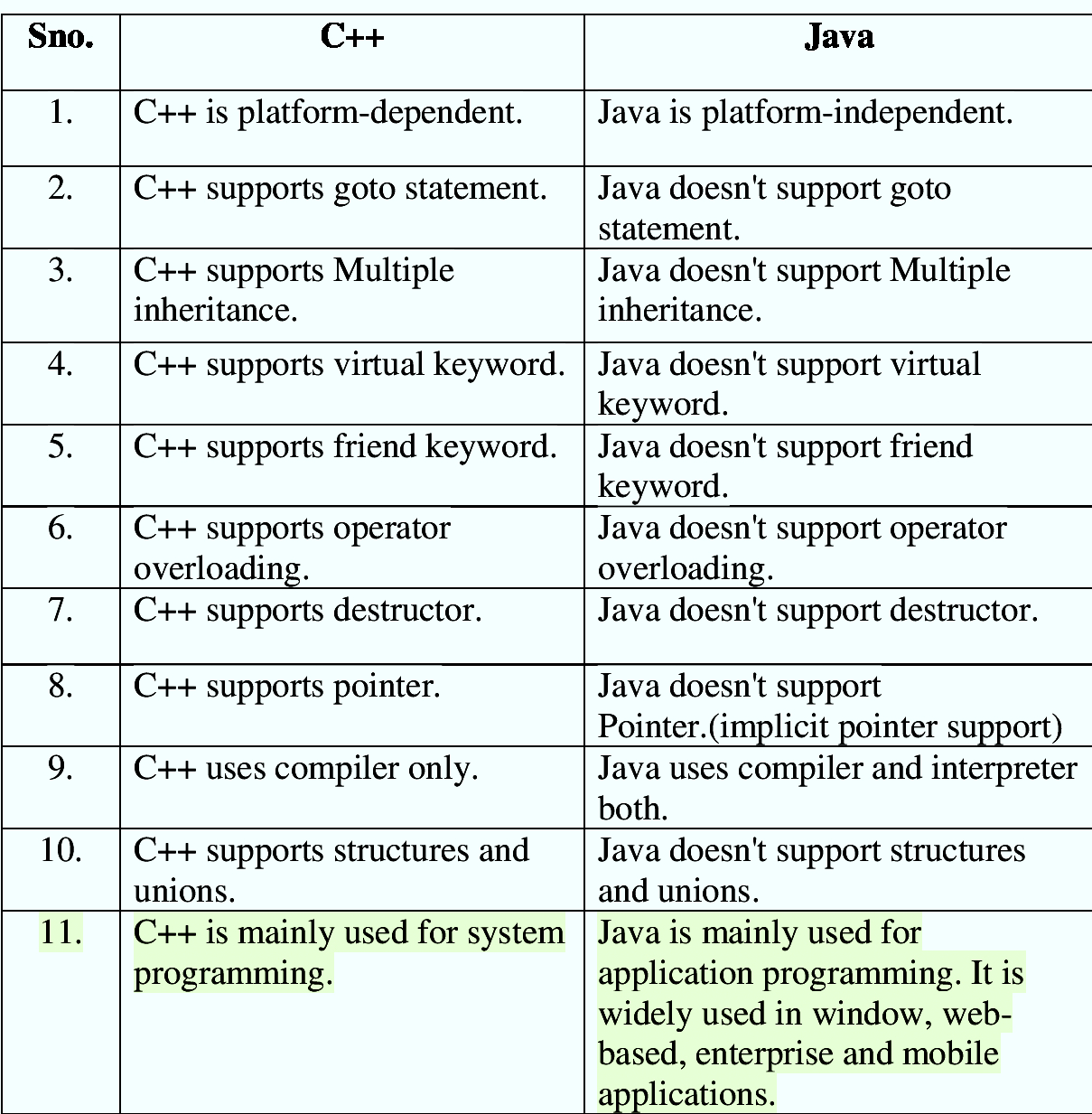
**Simple:**

Java is designed to be easy to learn. If you understand the basic concept of OOP Java, it would be easy to master.

**Secure:**

With Java's secure feature it enables to develop virus-free, tamper-free systems. Authentication techniques are based on public-key encryption.

**Difference between java and c++:**



**JVM:**

A Java virtual machine (JVM) is a virtual machine that enables a computer to run Java programs as well as programs written in other languages that are also compiled to Java bytecode. The JVM is detailed by a specification that formally describes what is required in a JVM implementation. Having a specification ensures interoperability of Java programs across different implementations so that program authors using the Java Development Kit (JDK) need not worry about idiosyncrasies of the underlying hardware platform.

The JVM reference implementation is developed by the OpenJDK project as open source code and includes a JIT compiler called HotSpot. The commercially supported Java releases available from Oracle Corporation are based on the OpenJDK runtime. Eclipse OpenJ9 is another open source JVM for OpenJDK.

**NetBeans:**

NetBeans is an integrated development environment (IDE) for Java. NetBeans allows applications to be developed from a set of modular software components called modules. NetBeans runs on Windows, macOS, Linux and Solaris. In addition to Java development, it has extensions for other languages like PHP, C, C++, HTML5,[4] and JavaScript. Applications based on NetBeans, including the NetBeans IDE, can be extended by third party developers.

**Eclipse:**

Eclipse is an integrated development environment (IDE) used in computer programming, and in 2014 was the most widely used Java IDE in one website's poll.[6] It contains a base workspace and an extensible plug-in system for customizing the environment. Eclipse is written mostly in Java and its primary use is for developing Java applications, but it may also be used to develop applications in other programming languages via plug-ins, including Ada, ABAP, C, C++, C#, Clojure, COBOL, D, Erlang, Fortran, Groovy, Haskell, JavaScript, Julia,[7] Lasso, Lua, NATURAL, Perl, PHP, Prolog, Python, R, Ruby (including Ruby on Rails framework), Rust, Scala, and Scheme. It can also be used to develop documents with LaTeX (via a TeXlipse plug-in) and packages for the software Mathematica. Development environments include the Eclipse Java development tools (JDT) for Java and Scala, Eclipse CDT for C/C++, and Eclipse PDT for PHP, among others.lipse:

**JDK:**

The Java Development Kit (JDK) is an implementation of either one of the Java Platform, Standard Edition, Java Platform, Enterprise Edition, or Java Platform, Micro Edition platforms[1] released by Oracle Corporation in the form of a binary product aimed at Java developers on Solaris, Linux, macOS or Windows. The JDK includes a private JVM and a few other resources to finish the development of a Java Application.[2] Since the introduction of the Java platform, it has been by far the most widely used Software Development Kit (SDK).[citation needed] On 17 November 2006, Sun announced that they would release it under the GNU General Public License (GPL), thus making it free software. This happened in large part on 8 May 2007, when Sun contributed the source code to the OpenJDK.

**EXPERIMENT NO. 2**

AIM: To Find Factorial of a no.

CODE:

package com.chanpreet;

import java.util.Scanner;

public class Factorial {

public static void main(String[] args) {

int fact = 1, i = 1, n;

System.out.println("Find factorial of a no.");

Scanner sc = new Scanner(System.in);

System.out.print("\nEnter value of n:");

n = sc.nextInt();

sc.close();

try {

if (n < 0)

throw new Exception("Factorial of negative numbers is not defined");

else if (n == 0)

System.out.println(1);

else {

for (i = 1; i <= n; i++)

fact = fact \* i;

System.out.println("Factorial of " + n + " is :" + fact);

}

}

catch (Exception e) {

System.out.println(e.getMessage());

}

}

}

**EXPERIMENT NO. 3**

AIM: To Print Prime Nos. till n

CODE:

package com.chanpreet;

import java.util.Scanner;

public class PrimeTillN {

public static void main(String[] args) {

int n;

boolean flag=true;

System.out.println("Find Prime nos till N");

Scanner sc=new Scanner(System.in);

System.out.println("\nEnter N:");

n=sc.nextInt();

sc.close();

System.out.println("prime nos till "+ n +" are:");

for(int i=2;i<=n;i++){

flag=true;

for(int j=2;j<=i/2;j++){

if(i%j==0){

flag=false;

break;

}

}

if(flag)

System.out.print(i+" ");

}

}

}

**EXPERIMENT NO. 4**

AIM: To find average marks of N Students and calculate their percentages and assign grades accordingly,

CODE:

package com.chanpreet;

import java.util.Scanner;

public class Student {

public int roll;

public String name;

public double[] marks;

public double avg;

public char grade;

public double percent;

public static void main(String[] args) {

Scanner sc=new Scanner(System.in);

System.out.print("Enter no of students");

int n=sc.nextInt();

Student[] obj=new Student[n];

for(int i=0;i<obj.length;i++) {

obj[i]=new Student();

System.out.println("Enter details of Student"+(i+1));

obj[i].input();

obj[i].calc();

}

for(int i=0;i<obj.length;i++) {

System.out.println("\nDetails of Student "+(i+1));

System.out.println("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_");

obj[i].disp();

}

System.out.println("Success");

}

private void disp() {

System.out.println("Name:"+name);

System.out.println("Roll No:"+roll);

System.out.println("Average Marks:"+avg);

System.out.println("Percentage"+percent);

System.out.println("Grade:"+grade);

}

private void calc() {

for(int i=0;i<marks.length;i++) {

avg+=marks[i];

}

double maxMarks=marks.length\*100;

percent=(avg/maxMarks)\*100;

avg/=marks.length;

if(percent>90)

grade='O';

else if(percent>80)

grade='A';

else if(percent>70)

grade='B';

else if(percent>60)

grade='C';

else if(percent>50)

grade='D';

else

grade='E';

}

private void input(){

Scanner sc=new Scanner(System.in);

System.out.print("Roll No. :");

roll=sc.nextInt();

sc.nextLine();

System.out.print("Name:");

name=sc.nextLine();

marks=new double[5];

for(int i=0;i<5;i++) {

System.out.print("Marks in subject "+(i+1));

marks[i]=sc.nextDouble();

}

}

}

**EXPERIMENT NO. 5**

AIM: To find the Biggest of 3 given Integer Nos.

CODE:

package com.chanpreet;

import java.util.Scanner;

public class BiggestOf3Int {

public static void main(String[] args) {

int a,b,c;

try{

System.out.println("Find biggest of 3 integers");

Scanner sc=new Scanner(System.in);

System.out.println("\nEnter the 3 integers:");

a=sc.nextInt();

b=sc.nextInt();

c=sc.nextInt();

sc.close();

int max= a>b ? (a>c ? a:c):(b>c? b:c);

System.out.println("Maximum of "+a+","+b+" and "+c+" is:"+max);

}

catch(Exception e){

System.out.println(e);

}

}

}

**EXPERIMENT NO. 6**

AIM: To define a class, define its constructor, overload the constructor and instantiate its object.

CODE:

package com.chanpreet;

public class ConstructorOverloading {

private String a;

public ConstructorOverloading(String a) {

this.a = a;

System.out.println(this.a);

}

public ConstructorOverloading() {

System.out.println("Default Constructor");

}

public static void main(String[] args) {

System.out.println("Constructor Overloading");

ConstructorOverloading obj1=new ConstructorOverloading();

ConstructorOverloading obj2=new ConstructorOverloading("Parameterised Constructor");

}

}

**EXPERIMENT NO. 7**

AIM: Create a Java Program to implement Stack and Queue.

CODE:

package com.chanpreet;

import java.util.LinkedList;

import java.util.Queue;

import java.util.Stack;

public class StackQueues {

public static void main(String[] args) {

System.out.println("Implement Stack and Queues");

Stack<Integer> stack=new Stack<Integer>();

Queue<Integer> queue=new LinkedList<Integer>();

System.out.println("\nInserting elements to stack and queue");

for(int i=0;i<10;i++){

System.out.println("Adding "+i);

stack.push(i); //adding to stack

queue.add(i); //adding to queue

}

System.out.println("\nElements in stack are:");

while(!stack.isEmpty()){

System.out.print(stack.peek()+" ");

stack.pop();

}

System.out.println("\nElements in queue are:");

while(!queue.isEmpty()){

System.out.print(queue.poll()+" ");

} } }

**EXPERIMENT NO. 8**

AIM: To demonstrate use of nested class.

CODE:

package com.chanpreet.nestedClass;

public class NestedClass {

int a=10;

private String str="Something!";

static double b=100.00;

void disp(){

System.out.println("\n\nOuter class display");

System.out.println("a:"+a+" b:"+b+" string:"+str);

}

class InnerClass{

int c=1;

void disp(){

System.out.println("\nInner class display");

System.out.println("a:"+a+" b:"+b+" c:"+c+" string:"+str);

}

}

}

package com.chanpreet.nestedClass; //Now the Main class

public class Main {

public static void main(String[] args) {

System.out.println("Demonstrate Nested Classes");

NestedClass obj1=new NestedClass();

obj1.disp();

NestedClass.InnerClass obj2= obj1.new InnerClass();

obj2.disp();

} }

**EXPERIMENT NO. 9**

AIM: To print real solutions of a quadratic equation a(x\*x)+b(x)+c.

CODE:

package com.chanpreet;

import java.util.Scanner;

public class RootsOfEquation {

public static void main(String[] args) {

System.out.println("Roots of Equation\n");

double a,b,c,d;

try{

System.out.println("Enter values of a,b and c:");

Scanner sc=new Scanner(System.in);

a=sc.nextDouble();

b=sc.nextDouble();

c=sc.nextDouble();

sc.close();

d= Math.pow(b, 2)-4\*a\*c;

if(d<0)

throw new Exception("There are no real roots for the given equation");

else{

double root1= (-b+Math.sqrt(d))/(2\*a);

double root2= (-b-Math.sqrt(d))/(2\*a);

System.out.println("Roots of the equation are:"+root1+" "+root2);

}

}catch(Exception e){

System.out.println(e.getMessage());

}

}

}

**EXPERIMENT NO. 10**

AIM: To write a java program to print Fibonacci Series with and without recursion

CODE:

package com.chanpreet;

import java.util.Scanner;

public class Fibonacci {

static int f=0,s=1,t;

public static void main(String[] args) {

Scanner sc=new Scanner(System.in);

System.out.println("Fibonacci Series\n");

System.out.println("Enter N:");

int n=sc.nextInt();

sc.close();

System.out.println("Without Recursion");

fiboWithoutRec(n);

System.out.println("\nWith Recursion");

System.out.print(f+" "+s);

fiboUsingRec(n-2);

}

private static void fiboUsingRec(int n) {

if(n>0){

t = f + s;

f =s;

s = t;

System.out.print(" "+t);

fiboUsingRec(n-1);

}

}

private static void fiboWithoutRec(int n) {

int first=0,second=1,third;

System.out.print(first+" "+second);

for(int i=2;i<n;i++){

third=first+second;

System.out.print(" "+third);

first=second;

second=third;

}

}

}