1. **Develop a JAVA program to demonstrate the precedence and associativity among arithmetic operators. The program should also demonstrate how the default precedence can be overridden.**

**import java.util.Scanner;**

**public class OperatorPrecedence {**

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

**Scanner sc = new Scanner(System.*in*);**

**// Read input from user**

**System.*out*.print("Enter first number: ");**

**int a = sc.nextInt();**

**System.*out*.print("Enter second number: ");**

**int b = sc.nextInt();**

**System.*out*.print("Enter third number: ");**

**int c = sc.nextInt();**

**// Without parentheses: multiplication has higher precedence than addition**

**int result1 = a + b \* c;**

**// With parentheses: overrides default precedence**

**int result2 = (a + b) \* c;**

**// Demonstrating associativity (left to right for same precedence)**

**int result3 = a - b + c;**

**System.*out*.println("a + b \* c = " + result1); // multiplication first**

**System.*out*.println("(a + b) \* c = " + result2); // addition first due to parentheses**

**System.*out*.println("a - b + c = " + result3); // evaluated left to right**

**}**

**}**

**Output:**

**a + b \* c = 610**

**(a + b) \* c = 900**

**a - b + c = 20**

1. **Write a program to generate the multiplication tables of a range of numbers between m and n inclusive and m<n.**

**import java.util.Scanner;**

**public class Demo1 {**

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

**int i,j,m,n;**

**Scanner sc=new Scanner(System.*in*);**

**System.*out*.println("Enter Starting number");**

**m=sc.nextInt();**

**System.*out*.println("Enter Starting number");**

**n=sc.nextInt();**

**if(m<n)**

**{**

**for(i=m;i<=n;i++)**

**{**

**for(j=1;j<=10;j++)**

**{**

**System.*out*.println(i+"X"+j+"="+(i\*j));**

**}**

**System.*out*.println();**

**}**

**}**

**}**

**}**

**Output:**

**Enter Starting number**

**1**

**Enter Starting number**

**2**

**1X1=1**

**1X2=2**

**1X3=3**

**1X4=4**

**1X5=5**

**1X6=6**

**1X7=7**

**1X8=8**

**1X9=9**

**1X10=10**

**2X1=2**

**2X2=4**

**2X3=6**

**2X4=8**

**2X5=10**

**2X6=12**

**2X7=14**

**2X8=16**

**2X9=18**

**2X10=20**

1. **Write a JAVA program to define a class, define instance methods for setting and retrieving values of instance variables and instantiate its object.**

**public class instanceDemo {**

**String Name;**

**int age;**

**double percentage;**

**void set(String stuname,int stuage,double stupercentage)**

**{**

**Name=stuname;**

**age=stuage;**

**percentage=stupercentage;**

**}**

**String getName()**

**{**

**return Name;**

**}**

**int getRoll()**

**{**

**return age;**

**}**

**double getpercent()**

**{**

**return percentage;**

**}**

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

**{**

**instanceDemo ob=new instanceDemo();**

**ob.set("Sam",21,60);**

**System.*out*.println("Name="+ob.getName());**

**System.*out*.println("Age="+ob.getRoll());**

**System.*out*.println("Percentage="+ob.getpercent());**

**}**

**}**

**Output:**

**Name=Sam**

**Age=21**

**Percentage=60.0**

1. **Write a JAVA program to demonstrate static member data and static member methods**

**public** **class** staticDemo {

**static** **int** *count*;

**static** **void** display()

{

System.***out***.println("count value="+*count*);

}

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

{

*count*=10;

*display*();

}

}

Output : count value=10

1. **Write a JAVA program to validate a date. The program should accept day, month and year and it should report whether they form a valid date or not.**

**import java.util.Scanner;**

**public class DateValidator {**

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

**Scanner sc = new Scanner(System.*in*);**

**// Take input from user**

**System.*out*.print("Enter day: ");**

**int day = sc.nextInt();**

**System.*out*.print("Enter month: ");**

**int month = sc.nextInt();**

**System.*out*.print("Enter year: ");**

**int year = sc.nextInt();**

**boolean valid = true; // Assume date is valid**

**// Validate year**

**if ((year <= 0) || (month < 1 || month > 12))**

**valid = false;**

**else {**

**int maxDays = 31; // Default**

**// Set max days based on month**

**if (month == 2) {**

**// Check for leap year**

**if ((year % 4 == 0 && year % 100 != 0) || year % 400 == 0)**

**maxDays = 29;**

**else**

**maxDays = 28;**

**} else if (month == 4 || month == 6 || month == 9 || month == 11)**

**maxDays = 30;**

**// Check if day is valid**

**if (day < 1 || day > maxDays)**

**valid = false;**

**}**

**// Print result**

**if (valid) {**

**System.*out*.println("Valid date="+day+"-"+month+"-"+year);**

**} else {**

**System.*out*.println("Invalid date");**

**}**

**}**

**}**

**Output:**

Enter day: 12

Enter month: 07

Enter year: 1981

Valid date=12-7-1981

1. **Write a JAVA program to implement inheritance and demonstrate use of method overriding.**

**class Shape {**

**void area() {**

**System.out.println("Area of shape is undefined");**

**}**

**}**

**class Circle extends Shape {**

**void area() {**

**System.out.println("Area of Circle = pi\*r\*r");**

**}**

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

**Shape s = new Circle();**

**s.area();**

**}**

**}**

**Output:**

**Area of Circle = π × r × r**

**7.**Write a JAVA program to demonstrate String class and its methods.

**import** java.util.Scanner;

**public** **class** Message1 {

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

Scanner sc = **new** Scanner(System.***in***);

System.***out***.println("Enter string:");

String str = sc.nextLine();

System.***out***.println("compareTo = " + str.compareTo("hello"));

System.***out***.println("length = " + str.length());

System.***out***.println("Upper = " + str.toUpperCase());

System.***out***.println("Lower = " + str.toLowerCase());

System.***out***.println("trim = " + str.trim());

**if** (str.length() > 5) {

System.***out***.println("charAt(5) = " + str.charAt(5));

} **else** {

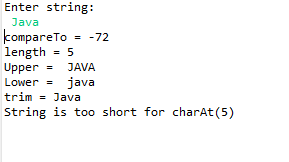
System.***out***.println("String is too short for charAt(5)");

}

}

}

**Output:**



8.Write a JAVA program to implement the concept of exception handling by creating user defined exceptions.

**class** ThrowDemo **extends** Exception

{

ThrowDemo(String str)

{

**super**(str);

}

}

**public** **class** ExeceptionDemo {

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

{

**try** {

**int** age=19;

**if**(age<18)

{

**throw** **new** ThrowDemo("Minor");

}

**else**

{

System.***out***.println("eligible to vote");

}

}

**catch**(ThrowDemo e)

{

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

}

}

Output:

Minor

9.Write a JAVA program to implement the concept of importing classes from user defined package and creating packages.

package mypack;

public class Addition

{

public int add(int a, int b)

{

return a + b;

}

}

package mypack;

public class Subtraction

{

public int subtract(int a, int b)

{

return a - b;

}

}

import mypack.Addition;

import mypack.Subtraction;

public class Test {

public static void main(String[] args) {

Addition ob1 = new Addition();

Subtraction ob2 = new Subtraction();

int a = 15, b = 10;

System.out.println("Addition of two no.=" + ob1.add(a, b));

System.out.println("Subtraction of two no. = " + ob2.subtract(a, b));

}

}

Output:

Addition of two no.=25

Subtraction of two no.=5

Write a JAVA program that creates three threads. First thread displays "Good Morning" every one second, the second thread displays "Hello" every two seconds and the third thread displays “Welcome” every three seconds.

**class** GoodMorningThread **extends** Thread {

**public** **void** run() {

System.***out***.println("Good Morning");

}

}

**class** HelloThread **extends** Thread {

**public** **void** run() {

System.***out***.println("Hello");

}

}

**class** WelcomeThread **extends** Thread {

**public** **void** run() {

System.***out***.println("Welcome");

}

}

**public** **class** MultiThreadExample {

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

{

**for** (**int** i = 0; i <3 ; i++) {

Thread t1 = **new** GoodMorningThread();

t1.start();

**try** { t1.join(); Thread.*sleep*(1000); } **catch** (Exception e) {}

Thread t2 = **new** HelloThread();

t2.start();

**try** { t2.join(); Thread.*sleep*(2000); } **catch** (Exception e) {}

Thread t3 = **new** WelcomeThread();

t3.start();

**try** { t3.join(); Thread.*sleep*(3000); } **catch** (Exception e) {}

}

}

}

Output:

Good Morning

Hello

Welcome

Good Morning

Hello

Welcome

Good Morning

Hello

Welcome