Index

Name: …… Bhalsod Aditya M.

Enrollment No: ……175690693001

Subject Name: ……..Programming in JAVA

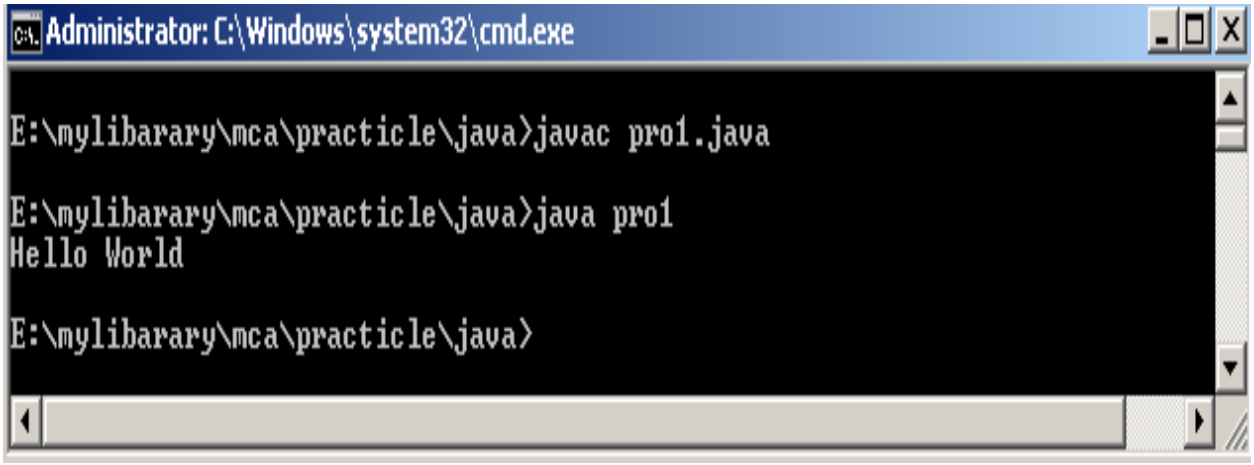
Subject Code:…… 4639302_Practical List

Semester: - ……iii

Program_List : ……..[1 To 21].

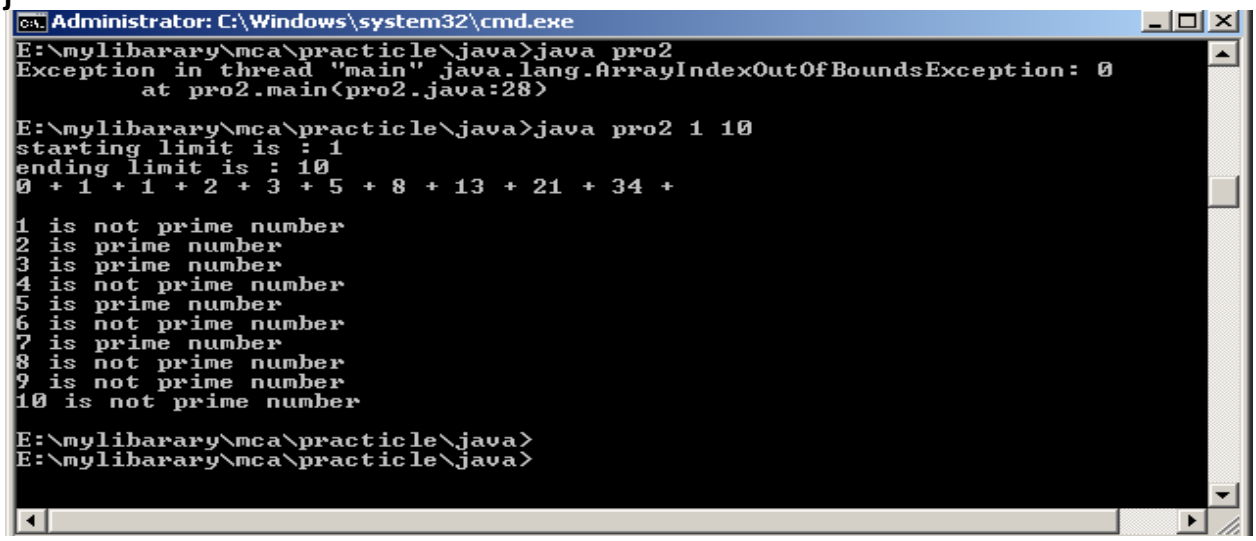
Pro-1) Write a simple “Hello World” java program, compilation, debugging, executing using java compiler and interpreter.

```
class pro1
{
        public static void main(String[] args) {
                System.out.println("Hello World");
        }
}
```



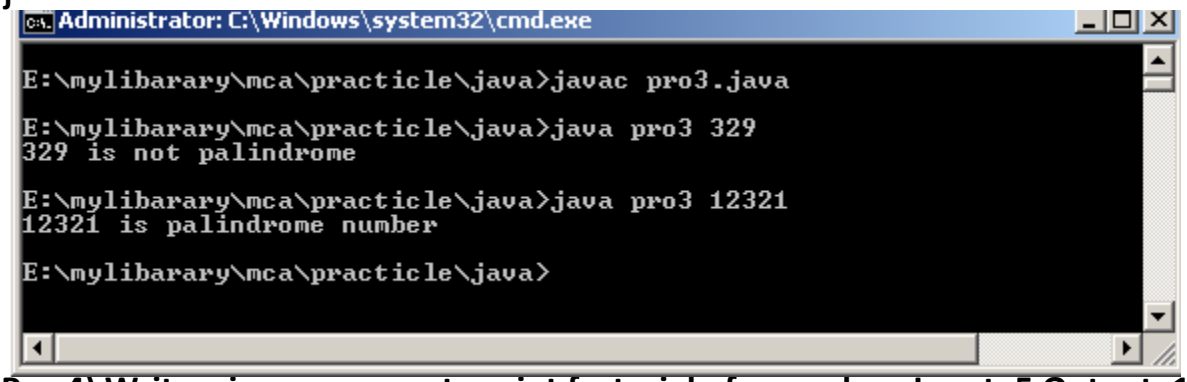
Pro-2) Write a program to pass Starting and Ending limit and print all prime numbers and Fibonacci numbers between this range.

```
class pro2{
        static void checkPrime(int n){
                //find prime number
                int k,m=0,flag=0;
                m=n/2;
                if(n==0||n==1){
                System.out.println(n+" is not prime number");
                }else{
                for(k=2;k<=m;k++){
                if(n%k==0){
                System.out.println(n+" is not prime number");
                flag=1;
                break;
                }
                }
                if(flag==0)  { System.out.println(n+" is prime number"); }
                }//end of else
        }
        public static void main(String[] args) {
        int t1=0,t2,l,s,e;
      s=Integer.parseInt(args[0]);
      e=Integer.parseInt(args[1]);
      t2=s;
      System.out.println("starting limit is : "+s);
     System.out.println("ending limit is : "+e);
for (int i=s; i <=e; ++i)//this for print fibonacci series
      {
         System.out.print(t1 + " + ");
         int sum = t1 + t2;
         t1 = t2;
         t2 = sum;
      }
      System.out.println("\n");
      for(l=s;l<=e;l++)
      {
                checkPrime(l);
      }
}
}
```



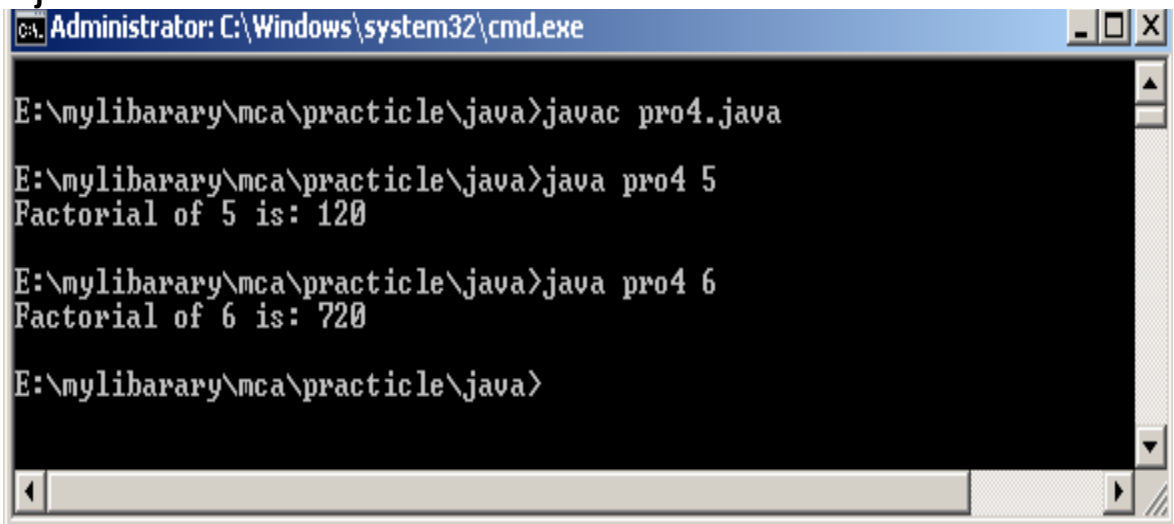
Pro-3) Write a java program to check palindrome number.Input: 329 Output: not palindrome number
Input: 12321 Output: palindrome number

```
class pro3
{
public static void main(String args[]){
     int r,sum=0,temp;
int n=Integer.parseInt(args[0]);
    temp=n;
     while(n>0){
      r=n%10;  //getting remainder
      sum=(sum*10)+r;
      n=n/10;
     }
     if(temp==sum)
      System.out.println(temp+" is palindrome number ");
     else
      System.out.println(temp+" is not palindrome");
   }
}
```



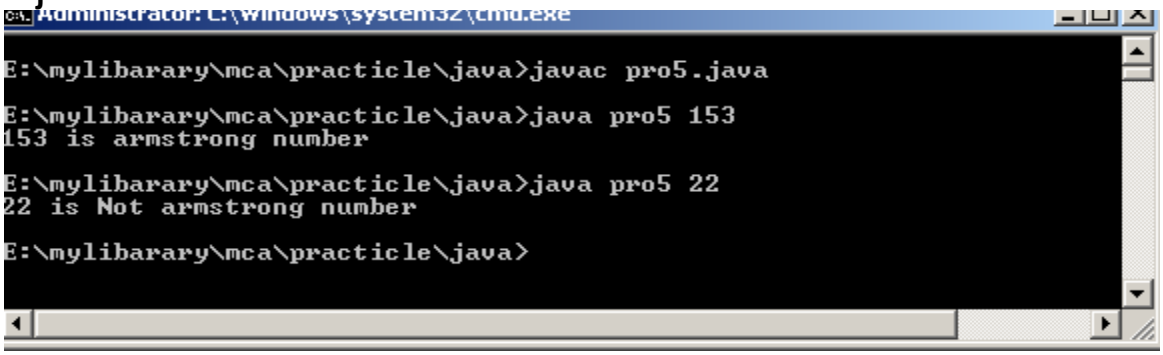
Pro-4) Write a java program to print factorial of a number .Input: 5 Output: 120. Input: 6 Output: 720.

```
   class pro4{
    public static void main(String args[]){
     int i,fact=1;
     int number=Integer.parseInt(args[0]);
     for(i=1;i<=number;i++){
        fact=fact*i;
     }
     System.out.println("Factorial of "+number+" is: "+fact);
    }
   }
```



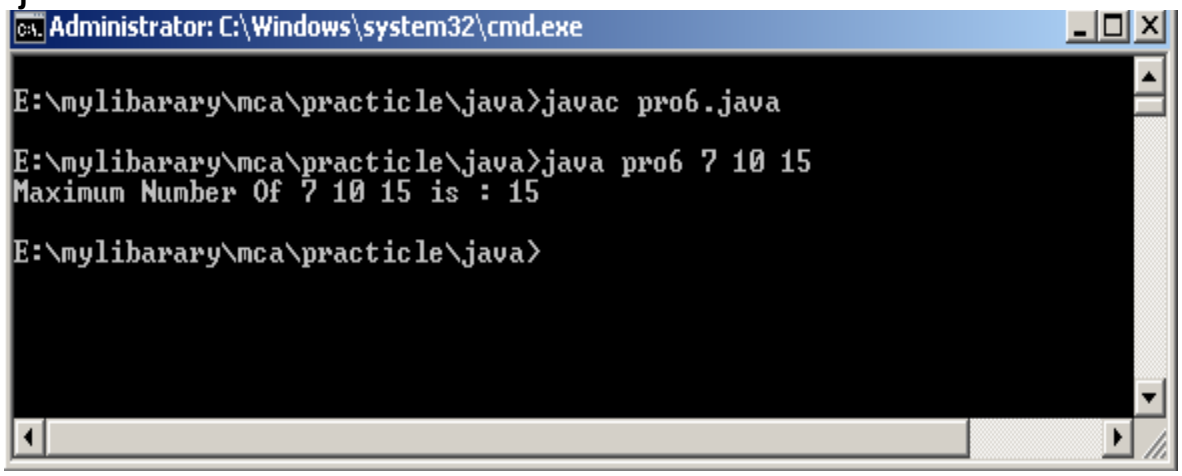
Pro-5) Write a java program to check Armstrong number.Input: 153 Output: Armstrong number.
Input: 22 Output: not Armstrong number.

```
class pro5{
  public static void main(String[] args)  {
    int c=0,a,temp;
    int n=Integer.parseInt(args[0]);
    temp=n;
    while(n>0)
    {
    a=n%10;
    n=n/10;
    c=c+(a*a*a);
    }
    if(temp==c)
    System.out.println(temp+" is armstrong number");
    else
       System.out.println(temp+" is Not armstrong number");
   }
}
```



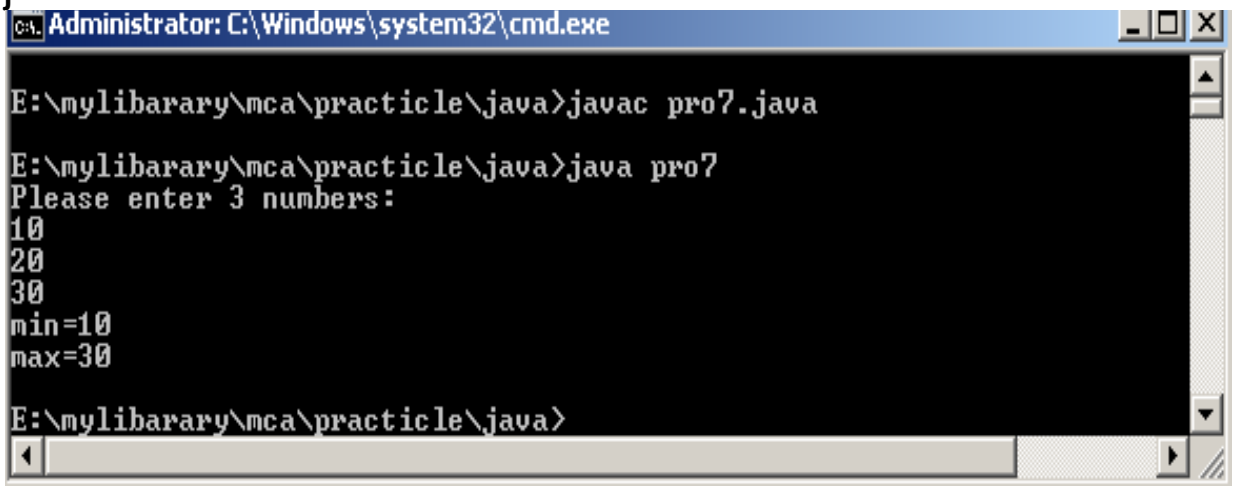
Pro-6) Write a program in Java to find maximum of three numbers using conditional operator.

```
class pro6{

    static int getLargest(int a,int b,int c){
      int largeest = (a>b?(a>c?a:c):(b>c?b:c));
      return largeest;
   }

   public static void main(String[] args) {
        int n1,n2,n3;

        n1=Integer.parseInt(args[0]);
        n2=Integer.parseInt(args[1]);
        n3=Integer.parseInt(args[2]);

        System.out.println("Maximum Number Of "+n1+" "+n2+" "+n3+" is :
"+getLargest(n1,n2,n3));
    }
 }
```



Pro-7) Write a java program which should display maximum and minimum number of given 3 numbers.

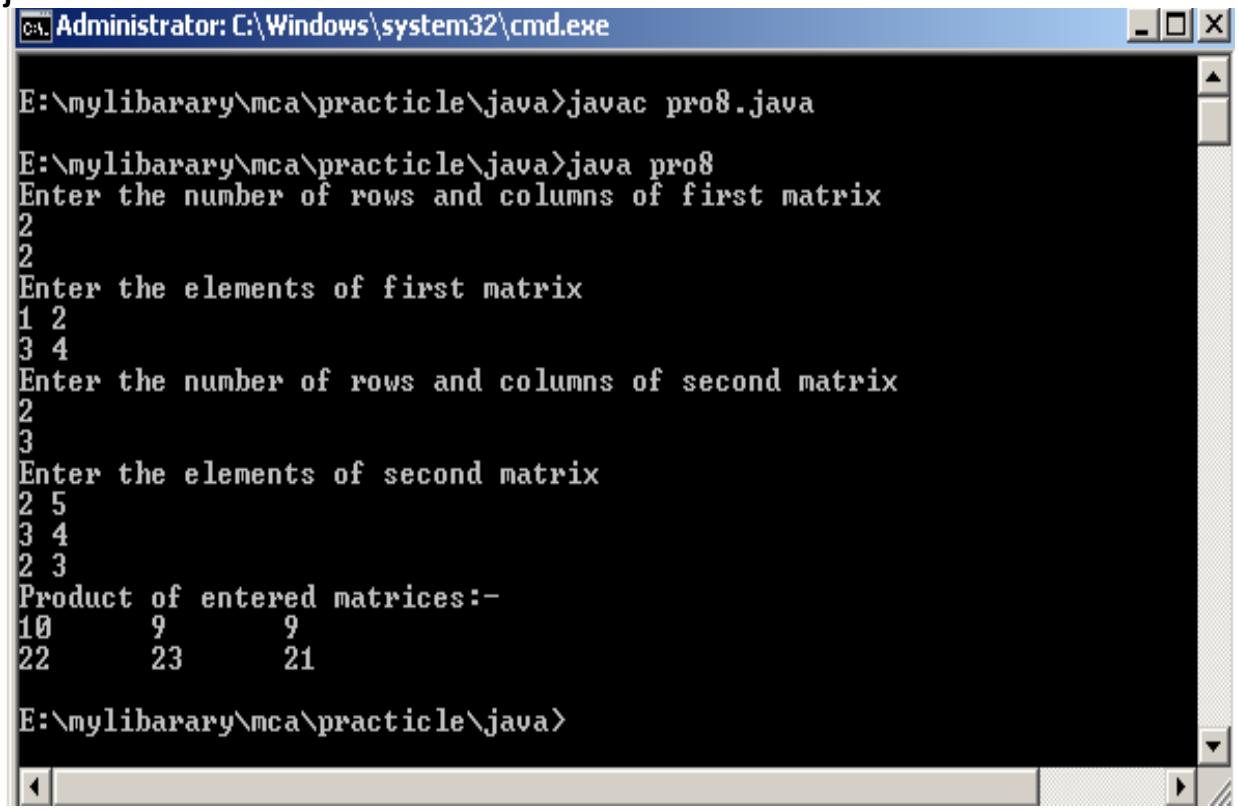
```
import java.util.Scanner;
class pro7{
public static void getMaxMin(int a, int b, int c){
      int max=a;
      int min=a;
      if(b>max){
         max=b;
      }
      if(c>max){
         max=c;
      }
      if(b<min){
         min=b;
      }
      if(c<min){
         min=c;
      }
      System.out.println("min="+min);
      System.out.println("max="+max);
   }
   public static void main(String[] args) {

      int [] numbers=new int[3];
      int n1,n2,n3;

        // create Scanner object
         Scanner input = new Scanner(System.in);

         // prompt user
         System.out.print("Please enter 3 numbers: \n");
         // use for loop to obtain user input
         for (int i= 0; i < numbers.length; i++) {
            numbers[i] = input.nextInt();
         } // end obtaining input
         n1=numbers[0];
         n2=numbers[1];
         n3=numbers[2];
         getMaxMin(n1,n2,n3);

   }
}
```



Pro-8) Write a program in Java to multiply two matrix.

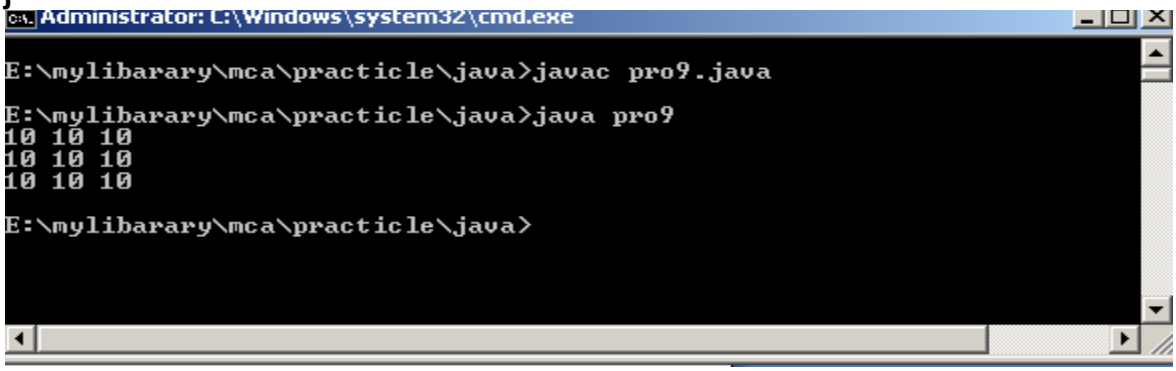
```
import java.util.Scanner;

class pro8
{
  public static void main(String args[])
  {
    int m, n, p, q, sum = 0, c, d, k;
    Scanner in = new Scanner(System.in);
    System.out.println("Enter the number of rows and columns of first matrix");
    m = in.nextInt();
    n = in.nextInt();
    int first[][] = new int[m][n];
    System.out.println("Enter the elements of first matrix");

    for ( c = 0 ; c < m ; c++ )
      for ( d = 0 ; d < n ; d++ )
        first[c][d] = in.nextInt();
    System.out.println("Enter the number of rows and columns of second matrix");
    p = in.nextInt();
    q = in.nextInt();

    if ( n != p )
      System.out.println("Matrices with entered orders can't be multiplied with each
other.");
    else
    {
      int second[][] = new int[p][q];
      int multiply[][] = new int[m][q];
      System.out.println("Enter the elements of second matrix");
     for ( c = 0 ; c < p ; c++ )
        for ( d = 0 ; d < q ; d++ )
          second[c][d] = in.nextInt();

      for ( c = 0 ; c < m ; c++ )
      {
        for ( d = 0 ; d < q ; d++ )
        {
          for ( k = 0 ; k < p ; k++ )
          {
            sum = sum + first[c][k]*second[k][d];
          }
          multiply[c][d] = sum;
          sum = 0;
        }
      }
      System.out.println("Product of entered matrices:-");




      for ( c = 0 ; c < m ; c++ )
      {

        for ( d = 0 ; d < q ; d++ )
          System.out.print(multiply[c][d]+"\t");
       System.out.print("\n");
      }
    }
  }
```

}


Administrator: C:\Windows\system32\cmd.exe
E:\mylibarary\mca\practicle\java>javac pro8.java
E:\mylibarary\mca\practicle\java>java pro8
Enter the number of rows and columns of first matrix
2
2
Enter the elements of first matrix
1 2
3 4
Enter the number of rows and columns of second matrix
2
3
Enter the elements of second matrix
2 5
3 4
2 3
Product of entered matrices:-
10 9 9
22 23 21
E:\mylibarary\mca\practicle\java>

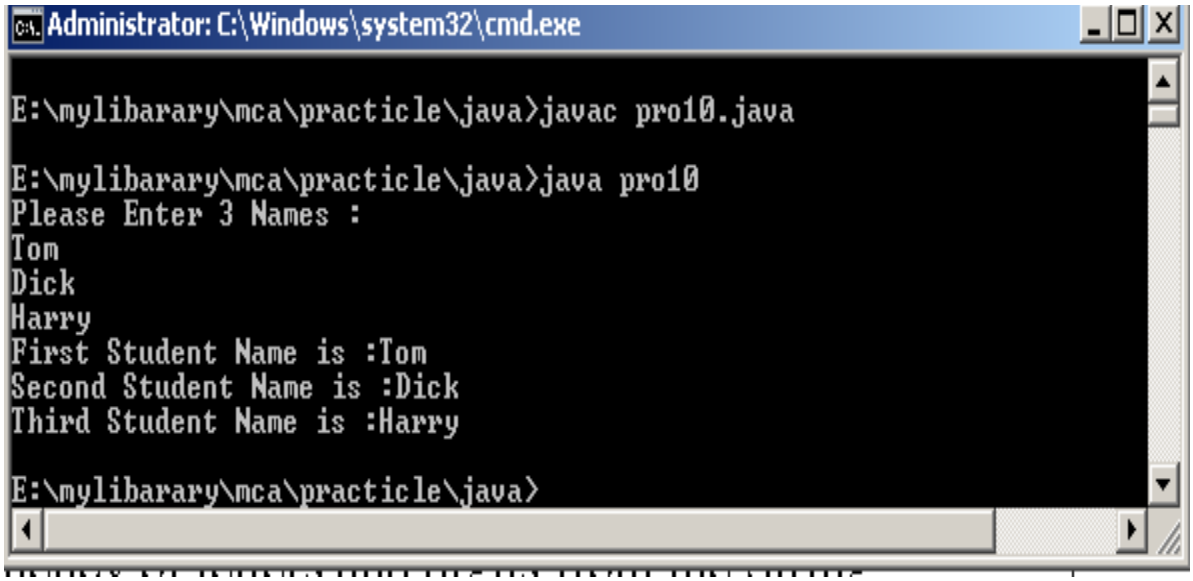
Pro-9) Write a java program to create a class “Matrix” that would contain integer values having varied numbers of columns for each row. Print row-wise sum of the integer values for each row.

```
public class pro9{

public static void main(String args[]){
//creating two matrices
int a[][]={{1,2,3},{4,5,6},{7,8,9}};
int b[][]={{9,8,7},{6,5,4},{3,2,1}};
//creating another matrix to store the sum of two matrices
int c[][]=new int[3][3];  //3 rows and 3 columns
//adding and printing addition of 2 matrices
for(int i=0;i<3;i++){
for(int j=0;j<3;j++){
c[i][j]=a[i][j]+b[i][j];    //use - for addition
System.out.print(c[i][j]+" ");
}
System.out.println();//new line
}
}
}
```



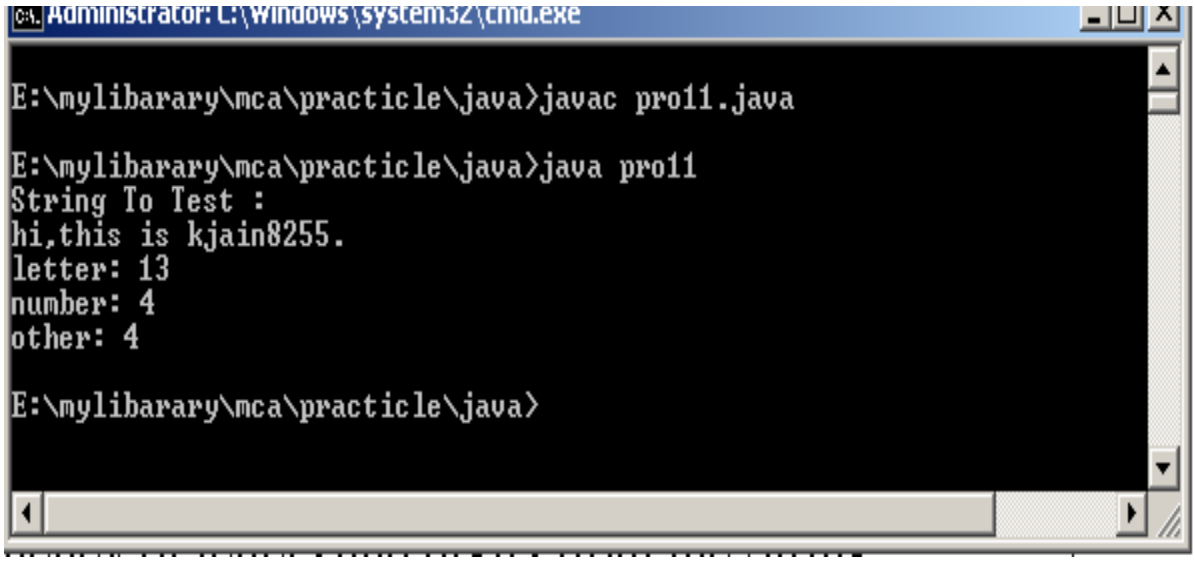
Pro-10) Write a Java application which takes several command line arguments, which are supposed to be names of students and prints output as given below:
(Suppose we enter 3 names then output should be as follows)..
Number of arguments = 3
1.: First Student Name is = Tom 2.: Second Student Name is = Dick 3.: Third Student Name is = Harry
Hint: An array may be used for converting from numeric values from 1 to 20 into String.

```
import java.util.Scanner;
class pro10{

        public static  void print_string(String s1,String s2,String s3)
        {
        System.out.println("First Student Name is :"+s1);
        System.out.println("Second Student Name is :"+s2);
        System.out.println("Third Student Name is :"+s3);

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

        String[] s= new String[3];
        String s2,s3,s4;
        Scanner input=new Scanner(System.in);
        System.out.print("Please Enter 3 Names : \n");

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

                s[i]=input.next();
        }

        s2=s[0];
        s3=s[1];
        s4=s[2];

        print_string(s2,s3,s4);

        }
}
```



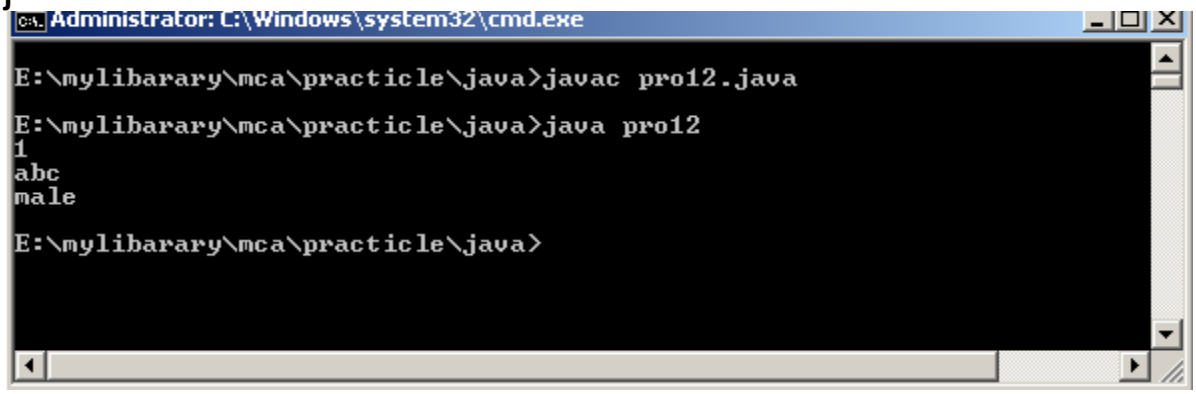
Pro-11) Write a Java application to count and display frequency of letters and digits from the String given by user as command-line argument.

```
import java.util.Scanner;
public class pro11{
public static void main(String[] args) {
		String test = "hi,this is kjain8255.";
		count(test);
}
	public static void count(String x){
		char[] ch = x.toCharArray();
		int letter = 0;
		int digit = 0;
		int other = 0;
		for(int i = 0; i < x.length(); i++){
			if(Character.isLetter(ch[i])){
				letter ++ ;
			}
			else if(Character.isDigit(ch[i])){
				digit ++ ;
			}
			else{
				other ++;
			}
		}

		System.out.println("String To Test :\nhi,this is kjain8255.");
		System.out.println("letter: " + letter);
		System.out.println("number: " + digit);
		System.out.println("other: " + other);
	}
}
```



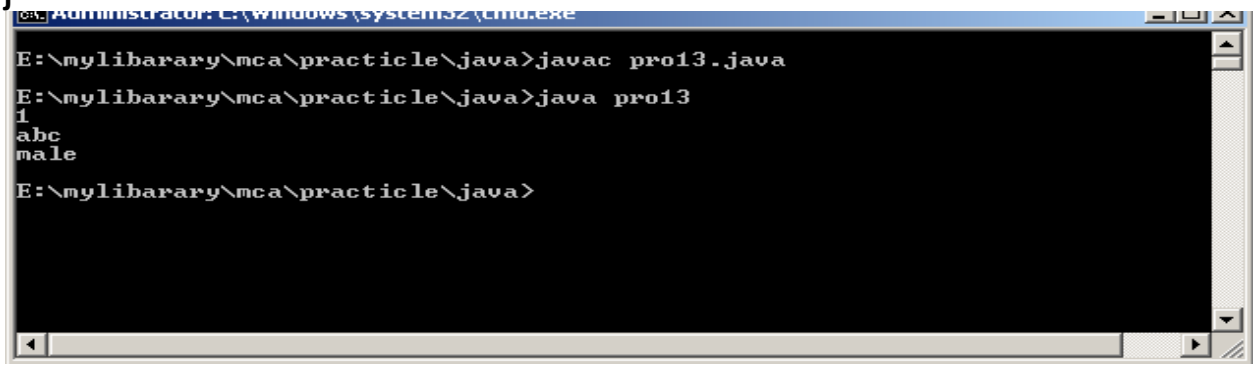
Pro-12) Create a class "Student" that would contain enrollmentNo, name, and gender as data members. Create appropriate getter and setter methods for the "Student" class and constructors to initialize the data members. Also demonstrate constructor chaining.

```
public class pro12
{
	private int no;
	String name;
	String gender;
	pro12(){
	no=0;
	}
	pro12(int no){
	this.no=no;
	}
	public void setNo(int no){
		this.no=no;
	}
	public void setName(String name){
		this.name=name;
	}
	public void setGender(String gender){
		this.gender=gender;
	}
	public void getNo(){
		System.out.println(no);
	}
	public void getName(){
		System.out.println(name);
	}
	public void getGender(){
		System.out.println(gender);
	}


	public static void main(String args[])
	{
		pro12 s1=new pro12();
		s1.setNo(1);
		s1.setName("abc");
		s1.setGender("male");
		s1.getNo();
		s1.getName();
		s1.getGender();
	}
}
```



Pro-13) Write a program in Java to demonstrate use of this keyword. Check whether this can access the private members of the class or not. [Refer class student in Q12 to perform the task]

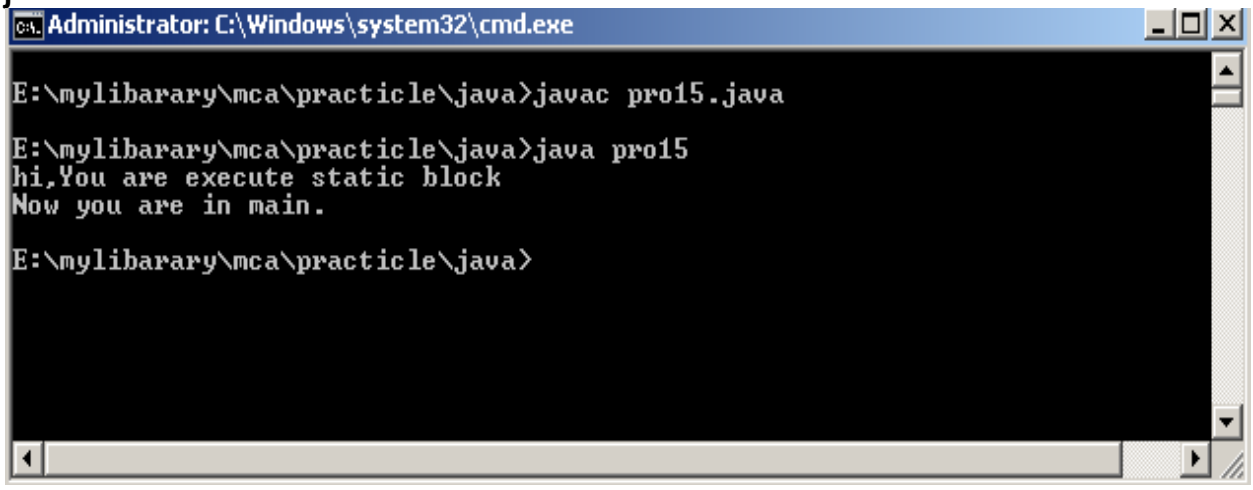
```
public class pro13
{
	private int no;
	String name;
	String gender;
	pro13()
	{
		no=0;
	}
	pro13(int no)
	{
		this.no=no;//this can access private member of class
	}
	public void setNo(int no)
	{
		this.no=no;
	}
	public void setName(String name)
	{
		this.name=name;
	}
	public void setGender(String gender)
	{
		this.gender=gender;
	}
	public void getNo(){
		System.out.println(no);
	}
	public void getName(){
		System.out.println(name);
	}
	public void getGender(){
		System.out.println(gender);
	}
	public static void main(String args[]){
		pro13 s1=new pro13();
		s1.setNo(1);
		s1.setName("abc");
		s1.setGender("male");
		s1.getNo();
		s1.getName();
		s1.getGender();
	}
}
```



Pro-14) Create a class “Rectangle” that would contain length and width as data members.

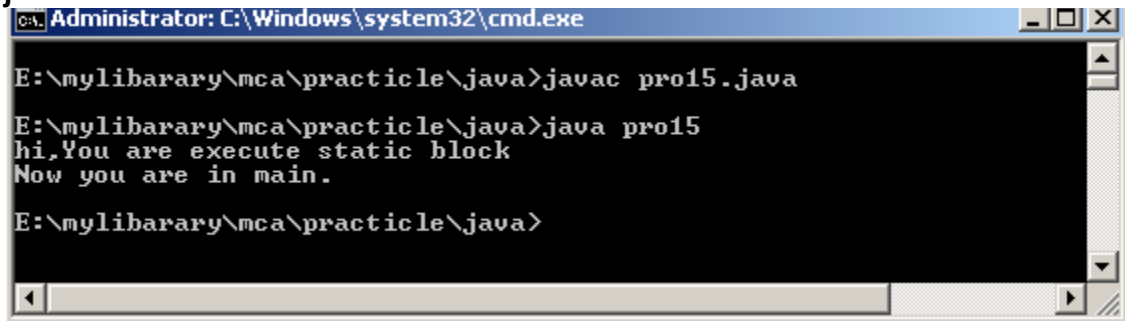
Define constructors [constructor overloading (default, parameterized and copy)] to initialize the data members. Define the member functions to find area and to display the number of objects created.
[Note: define initializer block, static initializer block and the static data member and member function. Also demonstrate the sequence of execution of initializer block and static initializer block]

```
class Rectangle{
        public static int length;
        public static int width;
        {
                System.out.println("This is initializer block");
        }
        Static
        {
                System.out.println("This is static initializer block");
        }
        public Rectangle() //default constructor
        {
                length=0;
                width=0;
        }

        public Rectangle(int length,int width) //parameterized constructor//constructor
overloading
        {
                this.length=length;
                this.width=width;
        }
        public static void area()
        {
                float area;
                area=length*width;
                System.out.println("Area of Rectangle is :"+area);
        }
}
public class pro14{

                public static void main(String[] args) {
                Rectangle rect=new Rectangle(10,20);
                rect.area();
        }
}
```



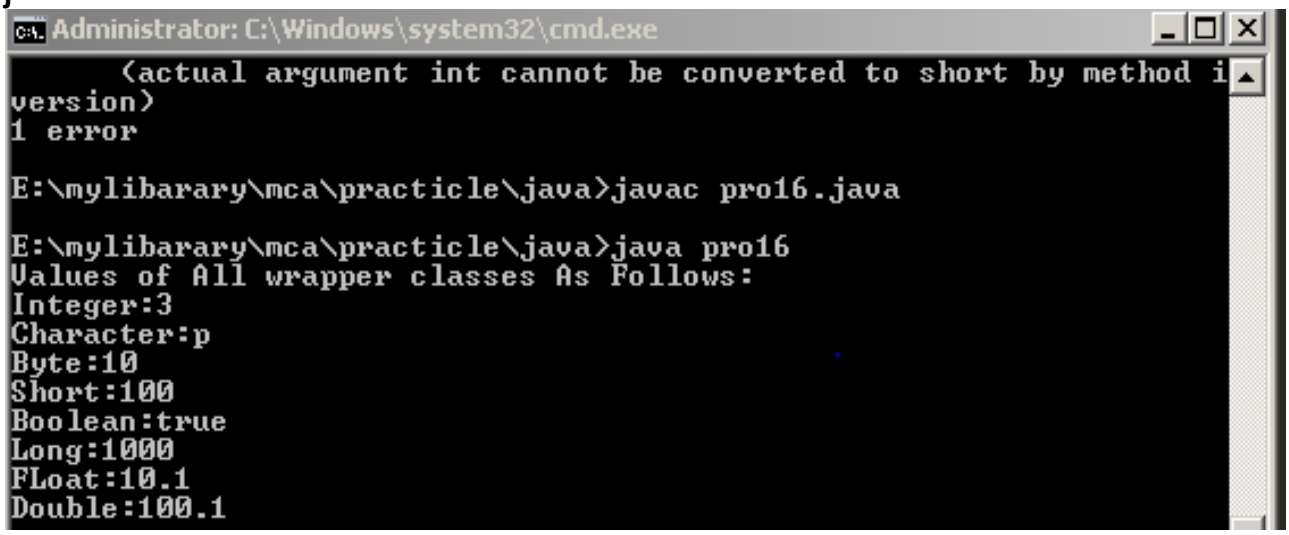
Pro-15) Write a java program static block which will be executed before main() method in class.

```
public class pro15{
  static
    {
       System.out.println("hi,You are execute static block");
    }
    public static void main(String[] args) {
       System.out.println("Now you are in main.");
    }
}
```



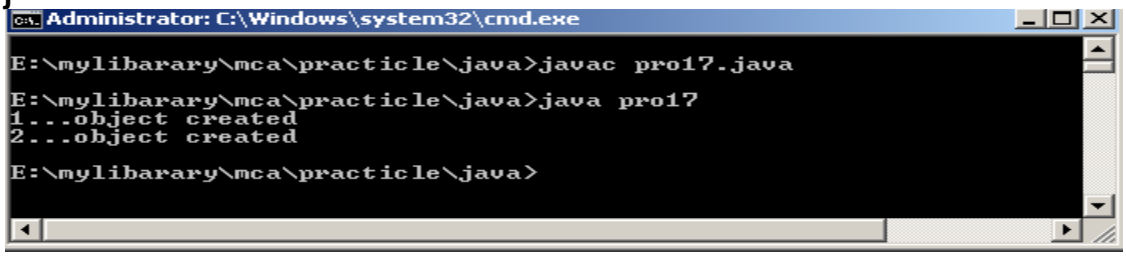
Pro-16) Write a programs in Java to use Wrapper class of each primitive data types.

```
public class pro16{
public static void main(String args[]){
        Integer i=new Integer(3);
        Character ch=new Character('p');
        Byte b=10;
        Short sh=100;
        Boolean bool=new Boolean(true);
        Long ln=new Long(1000);
        Float flt=new Float(10.10);
        Double dbl=new Double(100.100);
        System.out.println("Values of All wrapper classes As Follows:");
        System.out.println("Integer:"+i+"Character:"+ch+"Byte:"+b+"Short:"+sh+"Boolean:"+
bool+"Long:"+ln+"FLoat:"+flt+"Double:"+dbl);
}
}
```



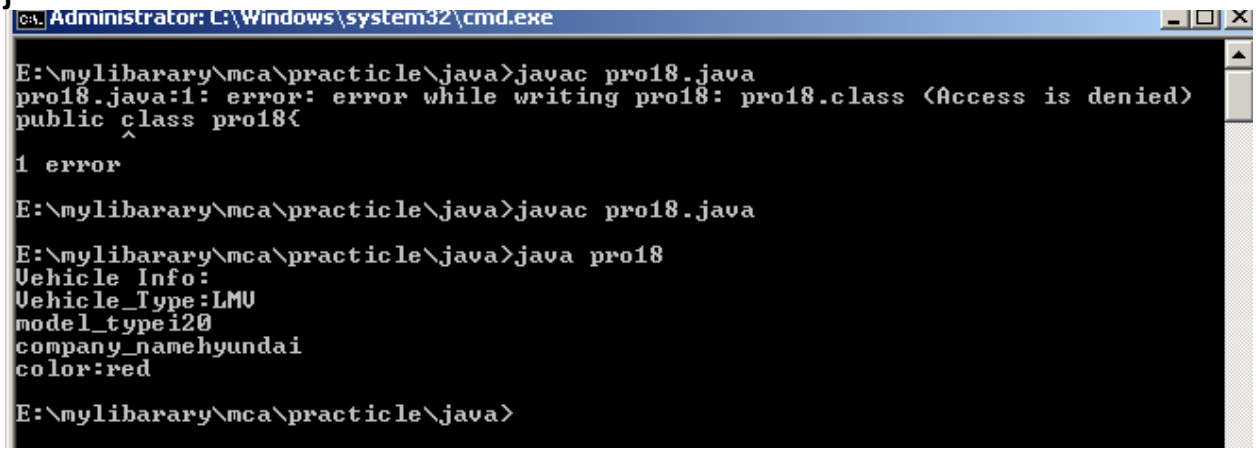
Pro-17) Write a class “circle” with radius as data member and count the number of instances created using default constructor only. [Constructor Chaining]

```
class pro17{
public static void main(String args[])
        {
                circle c1=new circle();
                circle c2=new circle();
        }
}
class circle
{
        static int count;
        static
        {
                count=0;
        }
        circle()
        {
                count++;
                System.out.println(count +"...object created");
        }
}
```



Pro-18) Create a class Vehicle with data member vehicle_type. Inherit the class in a class called car with data member model_type, company name etc. display the information of the vehicle by defining the display function in both super and sub class [Method Overriding]

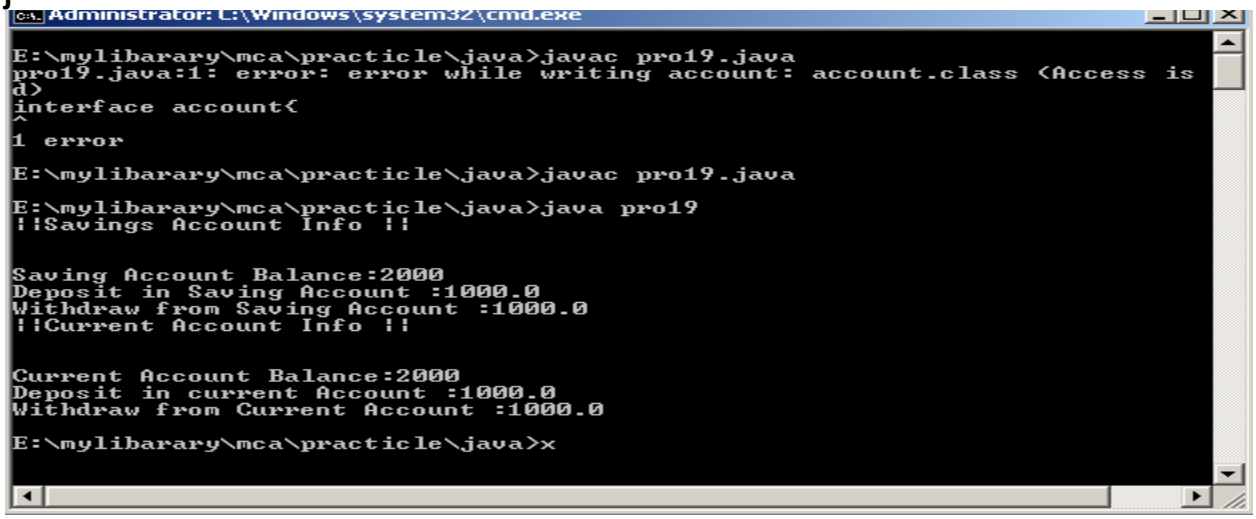
```
public class pro18{

	public static void main(String[] args) {

		Vehicle v=new Vehicle();
		Vehicle c=new car();
		c.display();
	}
}

class Vehicle{

	private String vehicle_type;

	public Vehicle(){
		vehicle_type="LMV";
	}

	public void display(){

		System.out.println("Vehicle Info:");
		System.out.println("Vehicle_Type:"+vehicle_type);
	};
}
class car extends Vehicle{

	public String model_type="i20";
	public String company_name="hyundai";
	public String color="red";
	public void display(){
		super.display();
	System.out.println("model_type"+model_type+"\ncompany_name"+company_name
+"\ncolor:"+color);
   }
}
```



Pro-19) Create a class “Account” containing accountNo, and balance as data members. Derive the Account class into two classes named “Savings” and “Current”. The “Savings” class should contain a data member named interestRate, and the “Current” class should contain a data member called overdraftLimit. Create appropriate member functions for all the classes to enable functionalities to check balance, deposit, and withdraw amount in Savings and Current account.
[Ensure that the Account class cannot be instantiated.]

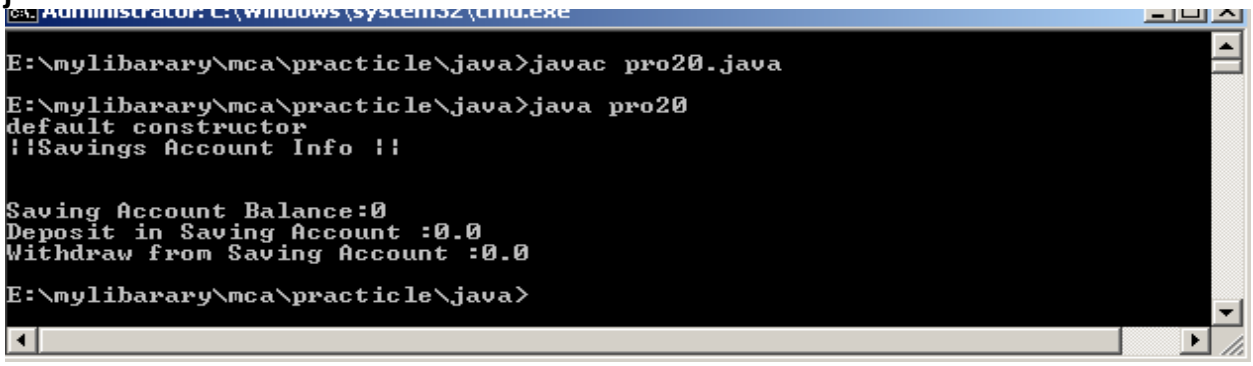
```
interface account{

	public String acoontno="70212010107772";
	public int balance=2000;
	public float deposit=1000;
	public float withdraw=1000;
}

class savings implements account{
		public float interest_rate=4;

	public void check_balance()
	{
		System.out.println("||Savings Account Info ||\n\n");
		System.out.println("Saving Account Balance:"+balance);
	}

	public void check_withdraw()
	{
		System.out.println("Withdraw from Saving Account :"+withdraw);
	}

	public void check_deposit()
	{

		System.out.println("Deposit in Saving Account :"+deposit);
	}


}

class current implements account{
	int overdraft_limit=10000;

	public void check_balance()
	{

		System.out.println("||Current Account Info ||\n\n");
		System.out.println("Current Account Balance:"+balance);
	}

	public void check_withdraw()
	{
		System.out.println("Withdraw from Current Account :"+withdraw);
	}

	public void check_deposit()
	{
		System.out.println("Deposit in current Account :"+deposit);
	}
}
public class pro19{
public static void main(String[] args) {
```

```
        savings s=new savings();
        current c =new current();
        s.check_balance();
        s.check_deposit();
        s.check_withdraw();
        c.check_balance();
        c.check_deposit();
        c.check_withdraw();
        }
}
```



Pro-20) Write a program in Java in which a subclass constructor invokes the constructor of the super class and instantiate the values.

```
public class pro20{
	public String acoontno="70212010107772";
	public int balance;
	public float deposit;
	public float withdraw;
	public pro20()
	{
		System.out.println("default constructor");
	}
	public pro20(int balance,float deposit,float withdraw)
	{
		this.balance=balance;
		this.deposit=deposit;
		this.withdraw=withdraw;
	}
public static void main(String[] args) {
	pro20 a=new pro20(2000,1000,1000);
	savings s=new savings();

	s.check_balance();
	s.check_deposit();
	s.check_withdraw();
	}


}
class savings extends pro20{
		public float interest_rate=4;
	public void check_balance()
	{
		System.out.println("||Savings Account Info ||\n\n");
		System.out.println("Saving Account Balance:"+balance);
	}
	public void check_withdraw()
	{
		System.out.println("Withdraw from Saving Account :"+withdraw);
	}

	public void check_deposit()
	{

		System.out.println("Deposit in Saving Account :"+deposit);
	}
}
```



Pro-21) Write a program in Java to demonstrate the use of 'final' keyword in the field declaration. How it is accessed using the objects.

```
//final variable can be initialized only once in a life
//we can not change its value that why its called constant.
public class pro21
{
	final int DAYS_IN_WEEK=7;
	int getdata()
	{
		//remove below comment and check
		//DAYS_IN_WEEK=8;
		return DAYS_IN_WEEK;
	}
	public static void main(String args[])
	{
		pro21 g=new pro21();
		System.out.println(g.getdata());
	}
}
```