**External Exam**

|  |  |
| --- | --- |
|  | Write a program to find maximum occurring character from given string |
| Ans: | package que\_1;  import java.util.Scanner;  public class Que\_1 {  public static void main(String[] args) {  Scanner sc=new Scanner(System.in);  String str = sc.next();  int[] freq = new int[str.length()];  char maxChar = str.charAt(0);  int i, j,max;    //Converts given string into character array  char string[] = str.toCharArray();    //Count each word in given string and store in array freq  for(i = 0; i < string.length; i++) {  freq[i] = 1;  for(j = i+1; j < string.length; j++) {  if(string[i] == string[j] && string[i] != ' ' && string[i] != '0') {  freq[i]++;    //Set string[j] to 0 to avoid printing visited character  string[j] = '0';  }  }  }  max=freq[0];  for(i=0;i<freq.length;i++)  {  if(max<freq[i])  {  max=freq[i];  maxChar=string[i];  }  }  System.out.println("Maximum occuring character :"+maxChar);  }  } |
|  | Write a program to print duplicate characters with number of occurrence in given string. |
| Ans: | package que\_2;  import java.util.Scanner;  class count  {  int c;  count(char str[])  {  for(int i = 0; i <str.length; i++) {  c = 1;  for(int j = i+1; j <str.length; j++) {  if(str[i] == str[j] && str[i] != ' ') {  c++;  //Set string[j] to 0 to avoid printing visited character  str[j] = '0';  }  }  //A character is considered as duplicate if count is greater than 1  if(c > 1 && str[i] != '0')  System.out.println("Duplicate word in given String is :"+str[i]);  }  }  }  public class Que\_2 {  public static void main(String[] args) {  Scanner sc=new Scanner(System.in);  String string1=sc.next();  //Converts given string into character array  char string[] = string1.toCharArray();  count c=new count(string);    }  } |
|  | Write a program to reverse the words of given string |
| Ans: | package que\_3;  public class Que\_3 {  public static void main(String[] args) {  // TODO code application logic here    String str= "my name is pritesh";  String a[]=str.split(" ");  for(int i=0;i<a.length;i++)  {  System.out.print(a[i]+" ");  }  System.out.println("");  for(int i=a.length-1;i>=0;i--)  {  System.out.print(a[i]+" ");  }  }  } |
|  | Write a java program to print from 1-15 and 15-1 using multithreading in java. |
| Ans: | package que\_4;  class multi extends Thread  {  public void run()  {  for(int i=1;i<=15;i++)  {  System.out.println(i);  try  {  Thread.sleep(1000);  }  catch(InterruptedException e)  {    }  }  }  }  class multi2 extends Thread  {  public void run()  {  for(int i=15;i>0;i--)  {  System.out.println(" "+ i);  try  {  Thread.sleep(1000);  }  catch(InterruptedException e)  {    }  }  }  }  public class Que\_4 {  public static void main(String[] args) {  multi t1=new multi();  multi2 t2=new multi2();  t1.start();  t2.start();  }  } |
|  |  |
|  | Create two thread, first thread will print prime number and second thread will print perfect number between 1-100. |
| Ans: | package que\_5;  class multi extends Thread  {  public void run()  {  int num=100,count = 0;  for(int i=1;i<=num;i++)  {  count=0;  for(int j=2;j<=i/2;j++)  {  if(i%j==0)  {  count++;  break;  }  }  if(count==0)  {  System.out.println(i);  }  try  {  Thread.sleep(500);  }  catch(InterruptedException e)  {    }  }    }  }  class multi2 extends Thread  {  public void run()  {    for(int i=1;i<=100;i++)  {  int sum=0;  for(int j=1;j<i;j++)  {  if(i%j==0)  sum+=j;  }  if(sum==i){  System.out.println( " "+i);}  try  {  Thread.sleep(500);  }  catch(InterruptedException e)  {    }  }  }  }  public class Que\_5 {  public static void main(String[] args) {  multi t1=new multi();  multi2 t2=new multi2();  t1.start();  t2.start();  }  } |
|  | Write program to accept rollno, marks of four subjects in an array from user and throw MarksOutOfBoundsException if marks are < 0 or marks > 100. Also check ArrayIndexOutofBoundsException. |
| Ans: | package que\_6;  import java.io.BufferedReader;  import java.io.InputStreamReader;  class MarksOutOfBoundException extends Exception  {  public String toString()  {  return "Enter marks in between 1 & 100";  }  }  class Student  {  BufferedReader br=new BufferedReader(new InputStreamReader(System.in));  int no;  double marks[];  Student()  {  no=0;  marks=new double[4];  for(int i=0;i<4;i++)  {  marks[i]=0;  }  }  public void accept()  {  System.out.println("Enter roll no & marks of 4 subjects ");  try  {  no=Integer.parseInt(br.readLine());  for(int i=0;i<4;i++)  {  marks[i]=Float.parseFloat(br.readLine());  if(marks[i]<0 || marks[i]>100)  throw new MarksOutOfBoundException();  }  }  catch(MarksOutOfBoundException me)  {  System.out.println(me);  }  catch(ArrayIndexOutOfBoundsException e)  {  System.out.println("Array index out of bounds");  }  catch(Exception ie)  {  System.out.println("Io Exception");  }  }  public void display()  {  System.out.println("Roll no\tMarks1\tMarks2\tMarks3\tMarks4");  System.out.print(no+"\t");  for(int i=0;i<4;i++)  System.out.print(marks[i]+"\t");  }  }  public class Que\_6 {  public static void main(String[] args) {  Student s1=new Student();  s1.accept();  s1.display();  }    } |
|  | Write a class Driver with attributes vehicle no, name & age. Initialize values through parameterized constructor. If age of driver is less than 18 then generate user-defined exception “AgeNotWithinTheRange” |
| Ans: | package que\_7;  import java.util.Scanner;  class AgeNotWithinTheRange extends Exception  {  AgeNotWithinTheRange(String s)  {  super(s);  }  }  class vehicle  {  void vehicle(int age) throws AgeNotWithinTheRange  {  if(age<18)  throw new AgeNotWithinTheRange("Age Not within the Range");  }  }  public class Que\_7 {  public static void main(String[] args) {  Scanner sc=new Scanner(System.in);  vehicle v=new vehicle();  try  {  String num =sc.next();  String name=sc.next();  int age=sc.nextInt();  v.vehicle(age);  }  catch(AgeNotWithinTheRange e)  {  System.out.println("Exception caught");  System.out.println(e.getMessage());  }  }  } |
|  |  |
|  | Code a java program in such a way that cover all the transition of library for issuing and submitting books for student as well as staff. For that follow the description.   1. There is one abstract class named **Lib** that contains three subjects in library like Java 100 books, SQL 150 books and CG 50 books. There are also two methods **submit()** and **issue()**. 2. There are two interfaces named **Student** and **Staff** contains one method   **info().** (Note: Both the interfaces has same name method info()).   1. There is one class named **LibTransitions** that calculate (for issue and submit) extends and implements appropriate classes and interfaces. 2. There is main class named **Library** that calls all methods according to user selection for Staff of Student.   Whenever transaction occurs stock must be updated for books. |
| Ans: | package que\_8;  import java.util.Scanner;  abstract class Lib  {  int java=100,SQL=150,CG=50;  abstract void submit();  abstract void issue();  }  interface student  {  void info();  }  interface staff  {  void info();  }  class LibTransection extends Lib  {  Scanner sc=new Scanner(System.in);  void submit() {  System.out.println("Which book would youn like issue to do?? java/SQL/CG");  String s=sc.next();  if(s.equalsIgnoreCase("java"))  {  if(java==100)  {  System.out.println("Invalid Submit for this book");  }  else  {  System.out.println("Book Submitted");  java++;  System.out.println("Stock :"+java);  }  }  if(s.equalsIgnoreCase("SQL"))  {  if(SQL==150)  {  System.out.println("Invalid Submit for this book");  }  else  {  System.out.println("Book Submitted");  SQL++;  System.out.println("Stock :"+SQL);  }  }  if(s.equalsIgnoreCase("CG"))  {  if(CG==50)  {  System.out.println("Invalid Submit for this book");  }  else  {  System.out.println("Book Submitted");  CG++;  System.out.println("Stock :"+CG);  }  } //To change body of generated methods, choose Tools | Templates.  }  void issue() {  System.out.println("Which book would youn like issue to do?? java/SQL/CG");  String s=sc.next();  if(s.equalsIgnoreCase("java"))  {  if(java==0)  {  System.out.println("No more Book.");  }  else  {  System.out.println("Book issued");  java--;  System.out.println("Stock :"+java);  }  }  if(s.equalsIgnoreCase("SQL"))  {  if(SQL==0)  {  System.out.println("No more Book.");  }  else  {  System.out.println("Book issued");  SQL--;  System.out.println("Stock :"+SQL);  }  }  if(s.equalsIgnoreCase("CG"))  {  if(CG==0)  {  System.out.println("No more Book.");  }  else  {  System.out.println("Book issued");  CG--;  System.out.println("Stock :"+CG);  }  }  //To change body of generated methods, choose Tools | Templates.  }  }  public class Que\_8 {  public static void main(String[] args) {  Scanner sc=new Scanner(System.in);  LibTransection lt=new LibTransection();  int ch=1;  while(ch!=0)  {  System.out.println("Select Number :");  System.out.println("0. Exit");  System.out.println("1.Issue ");  System.out.println("2. Submit");  ch=sc.nextInt();  switch(ch)  {  case 0:  break;  case 1:  lt.issue();  break;  case 2:  lt.submit();  break;  default:  System.out.println("Invalid number .");  }  }  }  } |
|  |  |
|  | Write a program that will define a superclass Shape that defines dim1 and dim2 and a method Area() for 2 dimensional shapes. Define rectangle, circle and triangle subclasses with overridden method Area() to calculate area of itself. |
| Ans: | package que\_9;  import java.util.Scanner;  class Shape  {  int dim1=5,dim2=6;  void Area()  {    }  }  class rect extends Shape  {  void Area()  {  double are=dim1\*dim2;  System.out.println("Rectagle area is :"+are);  }  }  class circle extends Shape  {  double pi=3.14,ar=0;  void Area()  {  ar=pi\*dim1\*dim1;  System.out.println("Circle Area is : "+ar);  }  }  public class Que\_9 {  public static void main(String[] args) {  Scanner sc=new Scanner(System.in);  rect obj=new rect();  circle c=new circle();  System.out.println("1. Rectagle ");  System.out.println("2. Circle");  System.out.println("Enter your choice:");  int ch=sc.nextInt();  switch(ch)  {  case 1:  obj.Area();  case 2:  c.Area();  }  }  } |
|  |  |
|  | Write a program that defines a class Student with name, rollno, marks1, marks2, marks3, total and percentage variables. Define constructor to initialize variables and a method to calculate total as well as percentage. Define a separate method displayData() to display all values. |
| Ans: | package que\_10;  import java.util.Scanner;  class calculate  {  String Sname;  int no, mark1,mark2,mark3,total;  double percentage;  calculate(String name,int rollno,int marks1,int marks2,int marks3)  {  Sname=name;  no=rollno;  mark1=marks1;  mark2=marks2;  mark3=marks3;  }  void cal()  {  total=mark1+mark2+mark3;  percentage=(mark1+mark2+mark3)/3;  }  void display()  {  System.out.println("Name :"+Sname);  System.out.println("Roll no :"+no);  System.out.println("Total :"+total);  System.out.println("Percentage :"+percentage);  }  }  public class Que\_10 {  public static void main(String[] args) {  Scanner sc=new Scanner(System.in);  String name=sc.next();  int rollno=sc.nextInt();  int marks1=sc.nextInt();  int marks2=sc.nextInt();  int marks3=sc.nextInt();    calculate c=new calculate(name,rollno,marks1,marks2,marks3);  c.cal();  c.display();  }  } |
|  |  |
|  | Define a class series with overloaded methods to print Fibonacci series. This class must contain the following type of method:  a. void fibo() : It will print 10 terms of series  b. void fibo(int terms): It will print total N terms specified by terms parameter.  void fibo(int start, int end) : It will print all the terms between starting and ending numbers |
| Ans: | package que\_11;  import java.util.Scanner;  class fibonacci  {  int n1,n2,n3,count;  void fibo()  {  n1=0;  n2=1;  count=10;  System.out.print(n1+" "+n2);  for(int i=2;i<count;++i)  {  n3=n1+n2;  System.out.print(" "+n3);  n1=n2;  n2=n3;  }  System.out.println("");  }  void fibo(int count)  {  n1=0;  n2=1;  System.out.print(n1+" "+n2);  for(int i=2;i<count;i++)  {  n3=n1+n2;  System.out.print(" "+n3);  n1=n2;  n2=n3;  }  System.out.println("");  }  void fibo(int start,int end)  {  int curr = 1, prev = 1;  while (curr <= end)  {  int temp = curr;  curr = prev + curr;  prev = temp;  if (curr >= start && curr <= end)  {  System.out.print(curr+" ");  }  }  }  }  public class Que\_11 {  public static void main(String[] args) {  fibonacci f=new fibonacci();  f.fibo();  f.fibo(11);  f.fibo(6,100);  }  } |
|  |  |
|  | Code a java program for the Oil and Refinery Company which receives transporting of raw materials. Consider there is one abstract class named **RowMaterial** having some instance variables like Goods-value(in Rs.) Service tax and Surcharge. There are two other sub classes like **ByRoad** and **ByShip.** There are two classes named **Railway** and **Transport** which follow the **ByRoad** class. Both these class have **calculate()** and **display()** methods. There is one class named **Ship** which follow **ByShip** and it also has **calculate()** and **display()** methods. There is a main class named **OilRef** where all these methods are  called according to user choice. Now you have to calculate service tax and surcharges according to Goods value. For this apply following conditions.   1. By Road:    * There is 12% service tax and 3% surcharge (on service tax) on goods value for both railway and transport. 2. By Ship:    * If transporting is out of country than service tax is 20% on goods value and 2% surcharge (on service tax).    * If transporting is within country than service tax is 10% on goods value and 2% surcharge (on service tax).   If user enter goods value less than 1 than generate custom exception for it. |
| Ans: |  |