

```

1 //UNIT-7 QB PROGRAMS
2 =====class and objects=====
3 // 492 Write a Java program to print "Hello LJU" message using class LJ.
4 class CO1{
5     void print(){
6         System.out.println("Hello");
7     }
8 }
9 class Run{
10     public static void main(String args[]){
11         CO1 ob = new CO1 ();
12         ob.print();
13     }
14 }
15 //493 Write a Java program to find area and perimeter of a circle using class
16 class CO2{
17     double area,r,peri;
18     double P = Math.PI;
19     void Area(double r){
20         area = P*r*r;
21         System.out.println("Area of circle = "+area);
22     }
23     void Perimeter(double r){
24         peri = 2*P*r;
25         System.out.println("Perimeter of circle = "+peri);
26     }
27 }
28 class Run{
29     public static void main(String args[]){
30         CO2 p1 = new CO2 ();
31         p1.Area(20);
32         p1.Perimeter(20);
33     }
34 }
35 //494 Write a Java program to count all digits of an integer number using class
36 import java.util.*;
37 class CO3{
38     int count=0;
39     Scanner sc = new Scanner(System.in);
40     void count(int n){
41         while(n!=0)
42         {
43             n=n/10;
44             count++;
45         }
46         System.out.println("Count of digit = "+count);
47     }
48 }
49 class Run{
50     public static void main(String args[]){
51         Scanner sc = new Scanner(System.in);
52         CO3 p1 = new CO3 ();
53         System.out.println("Enter any integer digit n:");
54         int n = sc.nextInt();
55         p1.count(n);

```

```

56     }
57 }
58 //495 Write a Java program to find product of all digits of an integer number using class
59 import java.util.*;
60 class CO4{
61     int sum = 0,r,product=1;
62     Scanner sc = new Scanner (System.in);
63     void sumPro(){
64         System.out.println("Eneter number n:");
65         int n = sc.nextInt();
66         while(n!=0)
67         {
68             r = n%10;
69             product=product*r;
70             n=n/10;
71         }
72         System.out.println("Product of all digit =" +product);
73     }
74 }
75 class Run{
76     public static void main(String args[]){
77         CO4 p1 = new CO4();
78         p1.sumPro();
79     }
80 }
81 //496 Write a Java program to check whether a given number is palindrome or not using
82 class
83 import java.util.*;
84 class CO5{
85     int temp=0,rev=0,r;
86     void peli(int n){
87         temp = n;
88         while(n!=0)
89         {
90             r=n%10;
91             rev = (rev*10) + r;
92             n=n/10;
93         }
94         if(temp == rev)
95         {
96             System.out.println("Yes Pelindrome");
97         }
98         else
99         {
100             System.out.println("Not Pelindrome");
101         }
102     }
103 }
104 class run{
105     public static void main(String args[]){
106         Scanner sc = new Scanner (System.in);
107         CO5 p1 = new CO5();
108         System.out.println("Enetr any integer number n");
109         int n = sc.nextInt();
110         p1.peli(n);

```

```

110     }
111 }
112 //497 Write a Java program to check whether a given number is Armstrong or not using
class
113 import java.util.*;
114 class C06{
115     double sum=0;
116     int r,count=0,temp=0;
117     void arm(int n){
118         temp = n;
119         while(n!=0)
120         {
121             n=n/10;
122             count++;
123         }
124         n=temp;
125         while(n!=0)
126         {
127             r=n%10;
128             sum= sum + ((Math.pow(r, count)));
129             n=n/10;
130         }
131         if(temp == sum)
132         {
133             System.out.println("Yes armstrong");
134         }
135         else
136         {
137             System.out.println("Not armstrong");
138         }
139     }
140 }
141 class run{
142     public static void main(String args[]){
143         Scanner sc = new Scanner (System.in);
144         C06 p1 = new C06();
145         System.out.println("Enetr any integer number n");
146         int n = sc.nextInt();
147         p1.arm(n);
148     }
149 }
150 //498 WAP to find nPr by using class and object
151 import java.util.*;
152 class Main_301{
153     public static void main(String[] arg){
154         Scanner sc = new Scanner(System.in);
155         System.out.print("Enter n: ");
156         int n = sc.nextInt();
157         System.out.print("Enter r: ");
158         int r = sc.nextInt();
159         Program p = new Program();
160         int Fact_N = p.factorial(n);
161         System.out.println("Factorial of n = "+Fact_N);
162         int Fact_N_R = p.factorial(n-r);
163         System.out.println("Factorial of n-r = "+Fact_N_R);

```

```

164         int nPr = (Fact_N)/(Fact_N_R);
165         System.out.println("Result of nCr = "+nPr);
166     }
167 }
168 class Program{
169     int factorial(int n)
170     {
171         int fact = 1;
172         for(int i=1 ; i<=n ; i++)
173         {
174             fact *= i;
175         }
176         return fact;
177     }
178 }
179
180 =====Methods and variables=====
181 //546 Create a class name Temperature in which create methods of given name ferenhit(),
182 celcius() to perform basic conversion.
183 //Call all this methods using class named Main.
184 class Temperature{
185
186     void ferenhit(double celsius){
187         double result = celsius * (9/5.0) + 32;
188         System.out.println("Temperature in fahrenheit is: "+result);
189     }
190     void celcius(double fahrenheit){
191         double result = (fahrenheit - 32) * (5/9.0);
192         System.out.println("Temperature in Celsius is: "+result);
193     }
194 }
195 class Main{
196     public static void main(String[] args) {
197         Temperature T1 = new Temperature();
198         T1.ferenhit(100);
199         T1.celcius(212);
200     }
201 }
202 //547 Write a java program to access two variables and two methods with different name
203 using class.
204 class Test {
205     int var1 = 10;
206     boolean var2 = true;
207     void method1(){
208         System.out.println("Method 1 Called");
209     }
210     void method2(){
211         System.out.println("Method 2 Called");
212     }
213 }
214 class Main{
215     public static void main(String[] args) {
216         Test T1 = new Test();
217         System.out.println("Variable 1 Accessed: "+T1.var1);
218         System.out.println("Variable 2 Accessed: "+T1.var2);

```

```

217         T1.method1();
218         T1.method2();
219     }
220 }
221 //548 Write a Java program to perform basic Calculator operations using following
methods catagories:
222 //sum() - without arguments without return
223 //minus() - with arguments without return
224 //multi() - without arguments with return
225 //divide() - with arguments with return
226 import java.util.*;
227 class EX311{
228     Scanner sc=new Scanner(System.in);
229     void sum() //without arguments without return,
230     {
231         System.out.println("Lets do Addition");
232         System.out.print("Enter value 1: ");
233         int a=sc.nextInt();
234         System.out.print("Enter value 2: ");
235         int b=sc.nextInt();
236         System.out.println("sum= "+(a+b));
237     }
238     void minus(int a, int b) //with arguments without return,
239     {
240         int c=a-b;
241         System.out.println("sub= "+c);
242     }
243
244     int multi() //without arguments with return
245     {
246         System.out.println("Lets do Multiplication");
247         System.out.print("Enter value 1: ");
248         int a=sc.nextInt();
249         System.out.print("Enter value 2: ");
250         int b=sc.nextInt();
251         int c=a*b;
252         return c;
253     }
254
255     double divide(int a,int b) //with arguments with return
256     {
257         double c= ((double)a/b);
258         return c;
259     }
260 }
261
262 class Run{
263     public static void main (String args[]){
264         Scanner sc=new Scanner(System.in);
265         EX311 e = new EX311();
266         e.sum();
267         System.out.println("Lets do Substraction");
268         System.out.print("Enter value 1: ");
269         int x=sc.nextInt();
270         System.out.print("Enter value 2: ");

```

```

271         int y=sc.nextInt();
272         e.minus(x,y);
273         int ans=e.multi();
274         System.out.println("multi= "+ans);
275
276         System.out.println("Lets do Division");
277         System.out.print("Enter value 1 : ");
278         int p=sc.nextInt();
279         System.out.print("Enter value 2 : ");
280         int q=sc.nextInt();
281         double ans2=e.divide(p,q);
282         System.out.println("division= "+ans2);
283     }
284 }
285 /*549 Design a class named Fan to represent a fan.
286 The class contains: - Three constants named SLOW, MEDIUM and FAST with values 1, 2 and 3
287 to denote the fan speed.
288 - An int data field named speed that specifies the speed of the fan (default SLOW).
289 - A boolean data field named f_on that specifies whether the fan is on (default false).
290 - A double data field named radius that specifies the radius of the fan (default 4).
291 - A data field named color that specifies the color of the fan (default blue).
292 - A no-arg method that creates a default fan.
293 - A parameterized method initializes the fan objects to given values.
294 - A method named display() will display description for the fan.
295 If the fan is on, the display() method displays speed, color and radius.
296 If the fan is not on, the method displays fan color and radius along with the message
297 "fan is off".
298 Write a test program that creates two Fan objects.
299 One with default values and the other with medium speed, radius 6, color brown, and
300 turned on status true.
301 Display the descriptions for two created Fan objects by comments.*/
302
303 class fann
304 {
305     int SLOW=1,MEDIUM=2,FAST=3;
306     int Speed;
307     boolean f_on;
308     double radius;
309     String color;
310
311     void defaultfan()
312     {
313         Speed=1;
314         f_on=false;
315         radius=4;
316         color="blue";
317     }
318
319     void fann(int a,boolean b,double c,String d)
320     {
321         Speed=a;
322         f_on=b;
323         radius=c;
324         color=d;
325     }

```

```

323
324     void display()
325     {
326         if(f_on==true)
327         {
328             System.out.println("Speed = "+Speed);
329             System.out.println("Radius = "+radius);
330             System.out.println("Color = "+color);
331
332         }
333         else
334         {
335             System.out.println("Fan is off");
336             System.out.println("Radius = "+radius);
337             System.out.println("Color = "+color);
338         }
339     }
340
341 }
342 class run1
343 {
344     public static void main(String args[])
345     {
346         fann f1 = new fann();
347         fann f2 = new fann();
348         f1.defaultfan();
349         f2.fann(2,true,6,"brown");
350         f1.display();
351         f2.display();
352     }
353 }
354 /*550 Define the Rectangle class that contains:
355 Two double fields x and y that specify the center of the rectangle, the data field width
356 and height,
357 A noarg method that creates the default rectangle with (0,0) for (x,y) and 1 for both
358 width and height.
359 A parameterized method creates a rectangle with the specified x,y,height and width.
360 -A method getArea() that returns the area of the rectangle.
361 -A method getPerimeter() that returns the perimeter of the rectangle.
362 -A method contains(double x, double y) that returns true if the specified point (x,y) is
363 inside this rectangle.
364 Write a test program that creates two rectangle objects. One with default values and
365 other with user specified values.
366 Test all the methods of the class for both the objects.*/
367
368 class rectangle{
369     double x,y,width,height;
370
371     void defaultData(){
372         x=0;
373         y=0;
374         width=1;
375         height=1;
376     }
377
378     void set(double a,double b,double c,double d){

```

```

374         x=a;
375         y=b;
376         width=c;
377         height=d;
378     }
379     void getArea(){
380         System.out.println("Area = "+(width*height));
381     }
382     void getPeri(){
383         System.out.println("Perimeter = "+(2*(width+height)));
384     }
385     boolean search(double a,double b){
386         if ( (a<=(x+(width/2))) && (a>=(x-(width/2))) && (b<=(y+(height/2))) && (b>=(y-(
height/2))) )
387         {
388             return true;
389         }
390         else
391         {
392             return false;
393         }
394     }
395 }
396 class run{
397     public static void main(String args[]){
398         rectangle r1 = new rectangle();
399         rectangle r2 = new rectangle();
400         r1.defaultData();
401         r2.set(0,0,2,2);
402         r1.getArea();
403         r1.getPeri();
404         r2.getPeri();
405         r2.getArea();
406         boolean q = r1.search(1.1,1);
407         System.out.println(q);
408     }
409 }
410 =====recursion=====
411 //571 Write a Java program to find HCF (Highest common factor) of two numbers using
recursion.
412 import java.util.*;
413 class GetHCF {
414     public static void main(String[] args) {
415         Scanner sc = new Scanner (System.in);
416         GetHCF g = new GetHCF();
417         System.out.print("Enter n1:");
418         int n1 = sc.nextInt();
419         System.out.print("Enter n2:");
420         int n2 = sc.nextInt();
421         int ans = g.hcf(n1, n2);
422         System.out.println("H.C.F of "+n1+ " and "+n2+ " is: "+ ans);
423     }
424     public int hcf(int n1, int n2){
425         if (n2 != 0)
426             return hcf(n2, n1 % n2);

```



```

427         else
428             return n1;
429     }
430 }
431
432 //572 Write a program to calculate factorial using recursion in Java
433
434 class RecursiveFactorial {
435
436     public static void main(String args[]) {
437         RecursiveFactorial rf = new RecursiveFactorial();
438         int fact = rf.factorial(5);
439         System.out.println(fact);
440     }
441
442     /* Java factorial program with recursion. */
443     int factorial(int num) {
444         if (num > 1) {
445             return num * factorial(num - 1);
446         }
447         else {
448             return 1;
449         }
450     }
451 }
452
453 //573 Write a Java program to calculate the power of a number like power (int number,
454 //int power) like power (2, 3) should return 8.
455 import java.util.*;
456 class Pow1 {
457     // Function to calculate N raised to the power P
458     int power(int N, int P){
459         if (P == 0)
460             return 1;
461         else
462             return N * power(N, P - 1);
463     }
464     public static void main(String[] args){
465         Pow1 p=new Pow1();
466         Scanner sc=new Scanner(System.in);
467         int N = sc.nextInt();
468         int P = sc.nextInt();
469         System.out.println(p.power(N, P));
470     }
471 }
472
473 //574 Write a Java program to convert Decimal to binary using recursion
474 import java.util.*;
475 class DtoB{
476     // Decimal to binary conversion
477     // using recursion
478     int find(int decimal_number){
479         if (decimal_number == 0)
480             return 0;
481         else
482             return (decimal_number % 2 + 10 *find(decimal_number / 2));
483     }
484 }

```

```

481 public static void main(String args[]){
482     DtoB d=new DtoB();
483     Scanner sc=new Scanner(System.in);
484     int decimal_number = sc.nextInt();
485     System.out.println(d.find(decimal_number));
486 }
487 }
488
489 //575 Write a Java program to find GCD and LCM of two numbers using recursion
490 import java.util.*;
491 class GcdLcm {
492     public static void main(String[] args) {
493         Scanner sc = new Scanner (System.in);
494         GcdLcm g = new GcdLcm();
495         System.out.print("Enter n1:");
496         int n1 = sc.nextInt();
497         System.out.print("Enter n2:");
498         int n2 = sc.nextInt();
499         int gcd = g.hcf( n1, n2);
500         System.out.println("G.C.D of "+n1+ " and "+n2+ " is: "+gcd);
501         System.out.printf("L.C.M of "+n1+ " and "+n2+ " is: "+((n1*n2)/gcd));
502     }
503     public int hcf(int n1, int n2){
504         if (n2 != 0)
505             return hcf(n2, n1 % n2);
506         else
507             return n1;
508     }
509 }
510 //576 Write a Java program to print fibonacci series upto n terms using recursion
511 class RecursiveFibo {
512     public static void main(String[] args) {
513         Scanner sc = new Scanner(System.in);
514         System.out.println("Please enter n:");
515         int n = sc.nextInt();
516         RecursiveFibo f = new RecursiveFibo();
517         for(int i=0;i<n;i++){
518             System.out.print(f.printFibo(i)+" ");
519         }
520     }
521     int printFibo(int count){
522         if(count==0)
523             return 0;
524         else if(count==1)
525             return 1;
526         else
527             return printFibo(count-1)+printFibo(count-2);
528     }
529 }
530 }
531 /*577 Declare array numArr = {7,32,64,2,10,23}
532 Find smallest element in this array using Recursion.
533 Output format : Smallest element in the array is: 2
534 (no need to display given array, no need of any user input class like scanner)*/
535 class SmallestElement {

```

```
536     public static void main(String[] args) {
537         SmallestElement s = new SmallestElement();
538         int[] numArr = {7, 32, 64, 2, 10, 23};
539         int smallest = s.findSmallest(numArr, numArr.length);
540         System.out.println("Smallest element in the array is: " + smallest);
541     }
542     int findSmallest(int[] arr, int n) {
543         if (n == 1) {
544             return arr[0];
545         }
546         return Math.min(arr[n - 1], findSmallest(arr, n - 1));
547     }
548 }
549
550 ===== // END OF CHAP-7 =====
551
552
```