

```

1 //UNIT-8 QB PROGRAMS
2 ===== method overloading =====
3 /*603 Write a program which asks user to choose one option to find area using method
overloading.
4 1. To calculate area of circle
5 2. To calculate area of rectangle
6 3. To calculate area of triangle
7 Methods to find area of circle,area of rectangle & area of triangle should be named as
"area". */
8
9 import java.util.*;
10 class OverloadDemo {
11     void area(double x) {
12         double z = 3.14 * x * x;
13         System.out.println("the area of the circle is " + z + " sq units");
14     }
15
16     void area(float x, float y) {
17         System.out.println("the area of the rectangle is " + x * y + " sq units");
18     }
19
20     void area(double base, double height) {
21         System.out.println("the area of the Triangle is " + 0.5 * base * height + " sq
units");
22     }
23
24 }
25
26 class Overload {
27     public static void main(String args[]) {
28         Scanner scanner = new Scanner(System.in);
29         OverloadDemo ob = new OverloadDemo();
30         System.out.println("Choose an option:");
31         System.out.println("1. To calculate area of circle");
32         System.out.println("2. To calculate area of rectangle");
33         System.out.println("3. To calculate area of triangle");
34         System.out.println("4. To Exit");
35         int choice;
36         do{
37             System.out.println("Enter your choice");
38             choice = scanner.nextInt();
39
40             if (choice == 1) {
41                 System.out.println("Enter radius");
42                 ob.area(scanner.nextDouble());
43             } else if (choice == 2) {
44                 System.out.println("Enter length and width of the rectangle");
45                 ob.area(scanner.nextFloat(),scanner.nextFloat());
46             } else if (choice == 3) {
47                 System.out.println("Enter base and height of the Triangle");
48                 ob.area(scanner.nextDouble(),scanner.nextDouble());
49             }
50         else{
51             System.exit(0);
52         }
53     }
54 }

```

```
53         }while(choice!=4);
54
55     }
56 }
57
58 /*604 Create a program that implements a calculator with overloaded methods for
59 addition, subtraction, multiplication, and division.
60 The calculator should have methods like calculate(int a, int b), calculate(double a,
61 double b), and so on to perform the respective operations based on the data types of the
62 input parameters.*/
63 class Calculator {
64
65     // Addition methods
66     int calculate(int a, int b) {
67         return a + b;
68     }
69
70     double calculate(double a, double b) {
71         return a + b;
72     }
73
74     // Subtraction methods
75     int calculateSubtraction(int a, int b) {
76         return a - b;
77     }
78
79     double calculateSubtraction(double a, double b) {
80         return a - b;
81     }
82
83     // Multiplication methods
84     int calculateMultiplication(int a, int b) {
85         return a * b;
86     }
87
88     double calculateMultiplication(double a, double b) {
89         return a * b;
90     }
91
92     // Division methods
93     int calculateDivision(int a, int b) {
94         if (b != 0) {
95             return a / b;
96         } else {
97             System.out.println("Error: Cannot divide by zero.");
98             return 0;
99         }
100    }
101
102    double calculateDivision(double a, double b) {
103        if (b != 0) {
104            return a / b;
105        } else {
106            System.out.println("Error: Cannot divide by zero.");
107            return 0.0;
108        }
109    }
110 }
```

```

105     }
106 }
107
108 public static void main(String[] args) {
109     Calculator calculator = new Calculator();
110
111     // Example usage
112     int resultInt = calculator.calculate(5, 3);
113     System.out.println("Result of integer addition: " + resultInt);
114
115     double resultDouble = calculator.calculate(5.5, 2.5);
116     System.out.println("Result of double addition: " + resultDouble);
117
118     int resultSubtract = calculator.calculateSubtraction(10, 4);
119     System.out.println("Result of integer subtraction: " + resultSubtract);
120
121     double resultSubtractDouble = calculator.calculateSubtraction(8.5, 3.0);
122     System.out.println("Result of double subtraction: " + resultSubtractDouble);
123
124     int resultMultiply = calculator.calculateMultiplication(6, 7);
125     System.out.println("Result of integer multiplication: " + resultMultiply);
126
127     double resultMultiplyDouble = calculator.calculateMultiplication(2.5, 4.0);
128     System.out.println("Result of double multiplication: " + resultMultiplyDouble);
129
130     int resultDivide = calculator.calculateDivision(12, 4);
131     System.out.println("Result of integer division: " + resultDivide);
132
133     double resultDivideDouble = calculator.calculateDivision(10.0, 2.0);
134     System.out.println("Result of double division: " + resultDivideDouble);
135
136     // Example of division by zero
137     double resultDivideByZero = calculator.calculateDivision(8.0, 0.0);
138     // The program will print an error message, and the result will be 0.0
139 }
140 }
141
142 /*605 Create a class comparison which has compare() method which compare two integer
143 value,
144 character value and double value using method overloading */
145 import java.util.Scanner;
146 class Compare
147 {
148     void compare(int a, int b) {
149
150         if (a == b) {
151             System.out.println("value of "+a+" and "+b+" are same");
152         }
153         else {
154             System.out.println("value of "+a+" and "+b+" are not same");
155         }
156     }
157
158     void compare(char a, char b) {

```

```

159         int x = (int)a;
160         int y = (int)b;
161
162         if (a == b) {
163             System.out.println("value of "+a+" and "+b+" are same");
164         }
165         else {
166             System.out.println("value of "+a+" and "+b+" are not same");
167         }
168     }
169
170     void compare(double a, double b) {
171
172         if (a == b) {
173             System.out.println("value of "+a+" and "+b+" are same");
174         }
175         else {
176             System.out.println("value of "+a+" and "+b+" are not same");
177         }
178     }
179
180 }
181
182 public static void main(String args[]) {
183     Scanner in = new Scanner(System.in);
184     Compare obj = new Compare();
185
186     System.out.print("Enter first integer: ");
187     int n1 = in.nextInt();
188     System.out.print("Enter second integer: ");
189     int n2 = in.nextInt();
190     obj.compare(n1, n2);
191
192     System.out.print("Enter first character: ");
193     char c1 = in.next().charAt(0);
194     System.out.print("Enter second character: ");
195     char c2 = in.next().charAt(0);
196     in.nextLine();
197     obj.compare(c1, c2);
198
199     System.out.print("Enter first double: ");
200     double s1 = in.nextDouble();
201     System.out.print("Enter second double: ");
202     double s2 = in.nextDouble();
203     obj.compare(s1, s2);
204 }
205 }
=====array of objects, passing arrays to methods=====
206 //622 Write a Java program to create an array of objects
207 import java.util.*;
208 class ArrayOf {
209     public static void main(String args[]) {
210         //create an array of product object
211         Product[] obj = new Product[5];
212         //create & initialize actual product objects using constructor

```

```

214     for(int i=0;i<5;i++) {
215         obj[i] = new Product();
216         obj[i].set();
217         obj[i].display();}
218     }
219 }
220
221 class Product {
222     int pro_Id;
223     String pro_name;
224     void set(){
225         Scanner sc=new Scanner(System.in);
226         pro_Id=sc.nextInt();
227         sc.nextLine();
228         pro_name=sc.next();
229     }
230     void display(){
231         System.out.print("Product Id = "+pro_Id + " " + " Product Name = "+pro_name);
232         System.out.println();
233     }
234 }
235
236 /*623 create a class Student with Roll_No ,Name and Mobile_No as data member.
237 Use necessary method to initialize it and to print. Create at least 5 student. (Use
array of object).*/
238 class Main{
239     public static void main(String[] arg){
240         Student a[] = new Student[5];
241         for(int i=0; i<5; i++)
242         {
243             a[i] = new Student();
244             System.out.println("----Enter Student "+(i+1)+" Data:----");
245             a[i].getData();
246         }
247         System.out.println("=====");
248         System.out.println("====Student Details====");
249         System.out.println("====");
250         for(int i=0; i<5; i++)
251         {
252             a[i].printData();
253         }
254     }
255 }
256 class Student
257 {
258     int RN;
259     String name;
260     long Mo_no;
261     void getData()
262     {
263         Scanner sc = new Scanner(System.in);
264         System.out.print("Enter Roll no: ");
265         RN = sc.nextInt();
266         sc.nextLine();
267         System.out.print("Enter Student name: ");

```

```

268         name = sc.nextLine();
269         System.out.print("Enter Mobile no: ");
270         Mo_no = sc.nextLong();
271         System.out.println();
272     }
273     void printData()
274     {
275         System.out.println();
276         System.out.println("Roll no= "+RN);
277         System.out.println("Student name= "+name);
278         System.out.println("Mobile No= "+Mo_no);
279     }
280 }
281
282 /*624 Write a Java Program to Create a class Account. It has three data member account
id, name and balance.
283 Define method to assign value and display value. Define method that search account
number given by the user.
284 If account number exists, print detail of that account. Write a program using array of
object.
285 Declare at least 5 account and print details.*/
286 import java.util.*;
287 class Account{
288     int id;
289     String name;
290     double balance;
291     Scanner sc =new Scanner(System.in);
292     void assign(){
293         System.out.println("Enter id no:");
294         id = sc.nextInt();
295         sc.nextLine();
296         System.out.println("Enter Name:");
297         name = sc.nextLine();
298         System.out.println("Enter balance:");
299         balance = sc.nextDouble();
300     }
301     void display(){
302         System.out.println("ID="+id);
303         System.out.println("Name="+name);
304         System.out.println("Balance="+balance);
305     }
306     int search(){
307         System.out.println("enter value of id you want to search x");
308         int x = sc.nextInt();
309         return x;
310     }
311 }
312 }
313 class Run
314 {
315     public static void main(String args[])
316     {
317         Account a[]=new Account[3];
318         for(int i=0;i<3;i++)
319         {

```

```

320         a[i]=new Account();
321         a[i].assign();
322         a[i].display();
323     }
324     Account s = new Account();
325
326     int x = s.search();
327
328     boolean flag= true; //to prevent multiple printing
329     for(int i=0;i<3;i++)
330     {
331         if(x==a[i].id)
332         {
333             a[i].display();
334             flag= false;
335             break;
336         }
337     }
338     if(flag==true)
339     {
340         System.out.println("Not found");
341     }
342 }
343 }
344
345 //625 Write a Java program to sort the numbers by using the concept of passing arrays to
methods
346 import java.util.*;
347 class PassArr{
348     void sort(int a[]){
349         int temp;
350         for(int i = 0; i<a.length; i++){
351             for(int j = i+1;j<a.length;j++){
352                 if(a[i]>a[j]){
353                     temp = a[i];
354                     a[i] = a[j];
355                     a[j] = temp;
356                 }
357             }
358         }
359     }
360     void display(int a[]){
361         for(int i = 0; i<a.length; i++){
362             System.out.print(a[i]+" ");
363         }
364     }
365
366
367 }
368 class Run{
369     public static void main(String args[]){
370         Scanner sc = new Scanner(System.in);
371         PassArr aa = new PassArr();
372         System.out.println("How many Elements:");
373         int n = sc.nextInt();

```

```

374     int a[] = new int[n];
375     System.out.println("Enter Elements:");
376     for(int i = 0; i<a.length; i++){
377         a[i] = sc.nextInt();
378     }
379     aa.sort(a);
380     System.out.println("After sorting Elements:");
381     aa.display(a);
382 }
383 }
384
385 //626 Write a Java program to display elements of one dimensional array using passing
386 arrays to methods.
387
388 class arrayDim{
389     void display(int[] array){
390         System.out.println("Your array is:");
391         for(int q: array)
392             System.out.print(q);
393     }
394     void display(int[][] array){
395         System.out.println("Your array is:");
396         for(int j[]: array)
397             for(int i: j)
398                 System.out.print(i);
399     }
400 }
401 class Run{
402     public static void main(String[] args) {
403         int[] a = {11,12,13};
404         int[][] b = {{1,2,3},{1,2,3},{1,2,3}};
405         arrayDim ad = new arrayDim();
406         ad.display(a);
407         ad.display(b);
408     }
409 }
410
411 //627 Write a Java program to find Even number and Odd number from given Array using the
412 concept of passing arrays to methods
413
414 class OddEven{
415     public static void main(String args[]){
416         int arr[]={1,2,5,6,3,2};
417         OddEven oe = new OddEven();
418         oe.oddprint(arr);
419         oe.evenprint(arr);
420     }
421     void oddprint(int a[]){
422         System.out.println("Odd Numbers:");
423         for(int i=0;i<a.length;i++){
424             if(a[i]%2!=0){
425                 System.out.println(a[i]);
426             }
427         }
428     }
429 }

```

```

427     void evenprint(int a[]){
428         System.out.println("Even Numbers:");
429         for(int i=0;i<a.length;i++){
430             if(a[i]%2==0){
431                 System.out.println(a[i]);
432             }
433         }
434     }
435 }
436
437 //628 Write a java program to find min and max values from a given array using passing
438 arrays to methods.
439 import java.util.Scanner;
440 class Array {
441     public int max(int[] array) {
442         int max = array[0];
443
444         for (int i = 0; i < array.length; i++) {
445             if (array[i] > max) {
446                 max = array[i];
447             }
448         }
449         return max;
450     }
451     public int min(int[] array) {
452         int min = array[0];
453         for (int i = 0; i < array.length; i++) {
454             if (array[i] < min) {
455                 min = array[i];
456             }
457         }
458         return min;
459     }
460     class Run
461     {
462         public static void main(String args[]) {
463             Scanner sc = new Scanner(System.in);
464             System.out.print("Enter the array range: ");
465             int n= sc.nextInt();
466             int[] arr = new int[n];
467             System.out.println("Enter the elements of the array: ");
468
469             for(int i=0; i< arr.length; i++) {
470                 arr[i] = sc.nextInt();
471             }
472             Array m = new Array();
473             System.out.println("Maximum value in the array is: "+m.max(arr));
474             System.out.println("Minimum value in the array is: "+m.min(arr));
475         }
476     }
477
478 //629 Write a java program to reverse elements of array using logic of swapping
479 elements using class. Here, use concept of passing array as argument to method. Use
480 Scanner class to enter Array elements.

```

```
479 import java.util.Scanner;
480 public class ReverseArray {
481
482     // Method to reverse the elements of the array using swapping logic
483     void reverseArray(int[] arr) {
484         int start = 0; //first index
485         int end = arr.length - 1; // last index
486
487         while (start < end) {
488             // Swap elements at start and end indices
489             int temp = arr[start];
490             arr[start] = arr[end];
491             arr[end] = temp;
492
493             // Move indices towards the center
494             start++;
495             end--;
496         }
497     }
498 }
499
500 class Run{
501     public static void main(String[] args) {
502         Scanner scanner = new Scanner(System.in);
503
504         // Input the size of the array
505         System.out.print("Enter the size of the array: ");
506         int size = scanner.nextInt();
507
508         // Input array elements
509         int[] array = new int[size];
510         System.out.println("Enter the array elements:");
511
512         for (int i = 0; i < size; i++) {
513             System.out.print("Element " + (i + 1) + ": ");
514             array[i] = scanner.nextInt();
515         }
516
517         // Display the original array
518         System.out.println("\nOriginal Array:");
519         for (int element : array) {
520             System.out.print(element + " ");
521         }
522
523         // Reverse the array
524
525         ReverseArray r = new ReverseArray();
526         r.reverseArray(array);
527
528         // Display the reversed array
529         System.out.println("\nReversed Array:");
530         for (int element : array) {
531             System.out.print(element + " ");
532         }
533 }
```

```

534     }
535 }
536
537 =====call by/pass by value/reference
538 =====
539 //636 Write a Java program to find sum of n numbers by using the concept of call by value
540 import java.util.*;
541 class Arr{
542     Scanner sc = new Scanner(System.in);
543     int sum(int n){
544         int sum=0;
545         for(int i = 1; i<= n; i++){
546             System.out.print("Enter elements: ");
547             int s = sc.nextInt();
548             sum = sum + s;
549         }
550         return sum;
551     }
552 }
553 class Run{
554     public static void main(String args[]){
555         Scanner sc = new Scanner(System.in);
556         Arr aa = new Arr();
557         System.out.println("How many Elements:");
558         int n = sc.nextInt();
559         System.out.println("Sum of "+n+ "numbers:"+aa.sum(n));
560     }
561 }
562 /*637 Write a Java program to add 5 in original value of array elements and display
modified elements of
one dimensional array using pass by value mechanism. */
563 class Array {
564     public static void main(String[] args) {
565         int a[] = {10, 20, 30, 40, 50};
566         Array obj = new Array();
567         obj.modifyArray(a);
568         for (int i = 0; i < a.length; i++) {
569             obj.print(a[i]);
570         }
571     }
572     void modifyArray(int a[]) {
573         for (int i = 0; i < a.length; i++) {
574             a[i] += 5;
575         }
576     }
577     void print(int a) {
578         System.out.println(a);
579     }
580 }
581 */
582 /*638 Write a Java program to change original value, which is given from user by using
appropriate Parameter passing technique. */
583 class CallByReference {
584     int a, b;
585     void set(int x, int y) {

```

```

586         a = x;
587         b = y;
588     }
589
590     void ChangeValue(CallByReference obj) {
591         obj.a += 10;
592         obj.b += 20;
593     }
594 }
595 public class Main {
596     public static void main(String[] args) {
597
598         CallByReference object= new CallByReference();
599         object.set(10, 20);
600         System.out.println("Value of a: " + object.a+ " & b: " + object.b);
601         object.ChangeValue(object);
602         System.out.println("Value of a: " + object.a+ " & b: " + object.b);
603     }
604 }
605
606 //639 Write a Java program to Swap two values, which is given from user by using call by
607 value
608 import java.util.Scanner;
609 class SwapNumber {
610     public static void main(String args[]) {
611         int x, y;
612         Scanner sc = new Scanner(System.in);
613         System.out.print("Enter the first number: ");
614         x = sc.nextInt();
615         System.out.print("Enter the second number: ");
616         y = sc.nextInt();
617         System.out.println("Before Calling in main\nx = "+x+"\ny = "+y);
618         SwapNumber s = new SwapNumber();
619         s.swap(x, y);
620         System.out.println("After calling in main\nx = "+x+"\ny = "+y); // no effect of
621         swap occurs in x,y here
622     }
623     void swap(int a, int b) {
624         a = a + b;
625         b = a - b;
626         a = a - b;
627         System.out.println("After Swapping inside method\nx = "+a+"\ny = "+b); // swap
628         occurs here only in a,b but not in original values x,y
629     }
630 }
631
632 //640 Write a Java program to Swap two values, which is given from user by using call by
633 reference.
634 class CallByReference {
635     int a, b;
636     void set(int x, int y) {
637         a = x;
638         b = y;
639     }
640     void swapvalue(CallByReference obj) {

```

```

637     int temp;
638     temp = obj.a;
639     obj.a = obj.b;
640     obj.b = temp;
641     System.out.println("After Swapping inside method\n Value of a: " + obj.a+ " & b:
642     " + obj.b); //swap occures in a,b of obj as well as in a,b of cbr
643 }
644 public class Main {
645     public static void main(String[] args) {
646
647         CallByReference cbr = new CallByReference();
648         cbr.set(10, 20);
649         System.out.println("Before Calling in main\n Value of a: " + cbr.a+ " & b: " +
650             cbr.b); // values set in instance of cbr.
651         cbr.swapvalue(cbr);
652         System.out.println("After calling in main\n Value of a: " + cbr.a+ " & b: " + cbr
653             .b); // effect of obj swapping occures in cbr also here
654     }
655
656 // 641 Write a program using call by reference concept to read an array of integers and
657 print its elements in reverse order
658 import java.util.Scanner;
659 class ReverseArray {
660
661     // Method to reverse the elements of the array using swapping logic
662     void reverseArray(int[] arr) {
663         int start = 0; //first index
664         int end = arr.length - 1; // last index
665
666         while (start < end) {
667             // Swap elements at start and end indices
668             int temp = arr[start];
669             arr[start] = arr[end];
670             arr[end] = temp;
671
672             // Move indices towards the center
673             start++;
674             end--;
675         }
676     }
677
678 class Run{
679     public static void main(String[] args) {
680         Scanner scanner = new Scanner(System.in);
681
682         int[] array = {2,5,4,8,9,1,7};
683
684         // Display the original array
685         System.out.println("\nOriginal Array:");
686         for (int element : array) {
687             System.out.print(element + " ");

```

```

688     }
689
690     // Reverse the array
691     ReverseArray r = new ReverseArray();
692     r.reverseArray(array);
693
694     // Display the reversed array
695     System.out.println("\nReversed Array:");
696     for (int element : array) {
697         System.out.print(element + " ");
698     }
699
700 }
701 }
702
703 //642 Write a Java program to copy one array into another array using call by
704 //reference.
705 class Array{
706     public static void main(String[] args) {
707         int a[] = {10,20,30,40,50};
708         int b[] = a;
709         Array obj = new Array();
710         obj.print(b);
711     }
712     void print(int a[]){
713         for (int i = 0; i<a.length;i++){
714             System.out.println(a[i]);
715         }
716     }
717 }
718 =====all concepts=====
719
720 /* 656 1. Program Objective:
721    The program's primary goal is to calculate the sum of two distances (in kilometers
722    and meters) and display the result in a user-readable format.
723 2. Classes and Methods:
724    - `Distance` class:
725        - `int km, m`: Instance variables to store kilometers and meters.
726        - `Scanner sc`: Used to take input from the user.
727        - `void get()`: Method to input distance values from the user.
728        - `void show(int a, int b)`: Method to display the distance in kilometers and
729          meters.
730        - `Distance add(Distance obj1, Distance obj2)`: Method to add two distances and
731          return the result as a new `Distance` object.
732    - `run` class:
733        - `main(String args[])`: The main method where the program execution begins.
734        - Creates instances of the `Distance` class, takes input for two distances,
735          calculates their sum, and displays the result.
736 3. Input and Output:
737    - The program prompts the user to input two distances: one in kilometers and another
738      in meters (or 0 if no meters are present).
739    - After input, it calculates the sum and displays it in kilometers and meters.
740 4. Distance Calculation:
741    - The `add` method of the `Distance` class takes two `Distance` objects, adds their

```

kilometer and meter values, and handles the carryover from meters to kilometers when necessary.

5. Output Format:

- The program displays the sum of distances in the format "Distance = X.Y KM," where X is kilometers, and Y is meters.*/

```
739
740     import java.util.*;
741 class Distance
742 {
743     int km,m;
744     Scanner sc = new Scanner(System.in);
745     void get()
746     {
747         System.out.println("Eneter value of distance in KM:");
748         km= sc.nextInt();
749
750         System.out.println("Eneter value of  distance in meter if any otherwise enter 0:");
751         );
752         m = sc.nextInt();
753     }
754
755     void show(int a,int b)
756     {
757         System.out.println("Distacne = "+a+"."+b+"KM");
758     }
759
760     Distance add(Distance obj1,Distance obj2)
761     {
762         Distance d4 = new Distance();
763         d4.km = obj1.km + obj2.km;
764         d4.m = obj1.m + obj2.m;
765
766         if(d4.m>=1000)
767         {
768             d4.km+=1;
769             d4.m-=1000;
770         }
771         return d4;
772     }
773 class run
774 {
775     public static void main(String args[])
776     {
777         Distance d1 = new Distance();
778         Distance d2 = new Distance();
779         Distance d3 = new Distance();
780         Distance d4 = new Distance();
781         d1.get();
782
783         d1.show(d1.km,d1.m);
784
785         d2.get();
786         d2.show(d2.km,d2.m);
787     }
```

```

788     d4 = d3.add(d1,d2);
789
790     d4.show(d4.km,d4.m);
791 }
792 }
793
794 /* 657 The provided Java code represents a program that deals with time calculations
795 using a `Time` class. Here's a problem definition for this code:
796
797 1. Program Objective:
798     The program's primary goal is to manipulate and display time values in terms of hours
799     and minutes.
800
801 2. Classes and Methods:
802     - `Time` class:
803         - `int hr, min`: Instance variables to store hours and minutes.
804         - `void Set(int x, int y)`: Method to set the time using provided hour and minute
805             values.
806         - `void Sum(Time obj1, Time obj2)`: Method to set the time by adding the hours and
807             minutes from two `Time` objects.
808         - `void print()`: Method to display the time in hours and minutes and handle the
809             conversion of excess minutes to hours if necessary.
810
811     - `TestA` class:
812         - `main(String args[])`: The main method where program execution begins.
813         - Creates instances of the `Time` class, sets time values, adds them, and displays
814             the result.
815
816 3. Time Manipulation:
817     - The `Time` class allows setting time using either direct hour and minute values or
818         by adding the hours and minutes from two `Time` objects.
819
820 4. Output Format:
821     - The program displays the time in hours and minutes in the format "hr=X" and
822         "min=Y," where X and Y represent the hour and minute values, respectively.*/
823
824
825 import java.util.*;
826 class Run {
827     public static void main(String[] args) {
828         Scanner sc= new Scanner(System.in);
829         System.out.println("Enter time-1 in hour");
830         int h1=sc.nextInt();
831         System.out.println("Enter time-1 in minutes");
832         int m1=sc.nextInt();
833         System.out.println("Enter time-2 in hour");
834         int h2=sc.nextInt();
835         System.out.println("Enter time-2 in minutes");
836         int m2=sc.nextInt();
837         Time t1= new Time();
838         Time t2= new Time();
839         Time t3= new Time();
840         t1.set(h1,m1);
841         t2.set(h2,m2);
842         t3.sum(t1,t2);
843         t3.print();
844     }
845 }
846
847 class Time
848 {

```

```

835     int hour;
836     int min;
837
838     void set(int h,int m)
839     {
840         hour=h;
841         min=m;
842     }
843
844     void sum(Time t1,Time t2)
845     {
846         min= t1.min+t2.min;
847         hour = (t1.hour+t2.hour)+(min/60);
848         min = min%60;
849     }
850
851     void print()
852     {
853         System.out.println("Total hours:"+hour);
854         System.out.println("Total minutes:"+min);
855     }
856 }
857
858 /* 658 Declare a class called coordinate to represent 3 dimensional Cartesian
coordinates(x, y, and z) define following method.
859 - Initialize Method
860 - Display method to print values of members
861 - Add_coordinates method, to add 3 such coordinates object to produce a resultant
coordinates object.
862 - Main , to show use of above method */
863 import java.util.*;
864 class Run {
865     public static void main(String[] args) {
866         Coordinate c1= new Coordinate();
867         Coordinate c2= new Coordinate();
868         Coordinate c3= new Coordinate();
869         Coordinate r= new Coordinate();
870
871         Scanner sc = new Scanner(System.in);
872         System.out.println("Enter first point's coordinates:");
873         int x1= sc.nextInt();
874         int y1= sc.nextInt();
875         int z1= sc.nextInt();
876         c1.input(x1,y1,z1);
877
878         System.out.println("\nEnter Second point's coordinates:");
879         int x2= sc.nextInt();
880         int y2= sc.nextInt();
881         int z2= sc.nextInt();
882         c2.input(x2,y2,z2);
883
884         System.out.println("\nEnter Third point's coordinates:");
885         int x3= sc.nextInt();
886         int y3= sc.nextInt();
887         int z3= sc.nextInt();

```

```

888         c3.input(x3,y3,z3);
889         c1.display();
890         c2.display();
891         c3.display();
892         Coordinate r1 = r.Add_coordinates(c1,c2,c3);
893
894         System.out.print("\nThe resultant coordinate is: "+r1.x+" "+r1.y+" "+r1.z);
895
896     }
897 }
898 class Coordinate
899 {
900     int x,y,z;
901     void input (int a, int b, int c)
902     {
903         x=a;
904         y=b;
905         z=c;
906     }
907
908     void display()
909     {
910         System.out.println("Coordinates of point is : (" + x + "," + y + "," + z + ")");
911     }
912
913     Coordinate Add_coordinates(Coordinate c1,Coordinate c2,Coordinate c3)
914     {
915         Coordinate c4 =new Coordinate();
916         c4.x=c1.x+c2.x+c3.x;
917         c4.y=c1.y+c2.y+c3.y;
918         c4.z=c1.z+c2.z+c3.z;
919         return c4;
920     }
921 }
922
923 //659 Write a Java program to find the power of number using passing an object to the
924 //method(i.e num=5 and power=3 then ans is 5^3 that is 125)
925 import java.util.*;
926 class Power{
927
928     int num,pow,result=1;
929
930     Power findPow(Power obj)
931     {
932         int i;
933         Power obj1=new Power();
934         for(i=1;i<=obj.pow;i++)
935             obj1.result=obj1.result*obj.num;
936     }
937     return obj1;
938 }
939
940 class run{

```

```

942 public static void main(String args[])
943 {
944     Power obj2=new Power();
945
946     Scanner sc=new Scanner(System.in);
947     System.out.println("Enter the Number ");
948     obj2.num=sc.nextInt();
949     System.out.println("Enter the value of Power");
950     obj2.pow=sc.nextInt();
951
952     Power obj3 =obj2.findPow(obj2);
953     System.out.println("power of given number is"+obj3.result);
954 }
955 }
956
957
958 //660 Write a Java program to find GCD of two numbers using passing an object to the
959 //method(i.e a=4 and b=6 then GCD is 2)
960 //FIRST LOGIC
961 import java.util.*;
962 class FindGCD
963 {
964     int a,b;
965     void input()
966     {
967         Scanner sc=new Scanner(System.in);
968         System.out.println("ENTER a and b=");
969         a=sc.nextInt();
970         b=sc.nextInt();
971     }
972     void gcd(FindGCD ob1)
973     {
974         while(ob1.a!=ob1.b)
975         {
976             if(ob1.a>=ob1.b) //when first num is bigger, assign in first
977                 ob1.a=ob1.a-ob1.b;
978             else //when second num is bigger, assign in second
979                 ob1.b=ob1.b-ob1.a;
980         }
981         System.out.println("GCD="+ob1.a);
982     }
983 class Run
984 {
985     public static void main(String args[])
986     {
987         FindGCD ob1 = new FindGCD();
988         FindGCD ob2 = new FindGCD();
989         ob1.input();
990         ob2.gcd(ob1);
991     }
992 }
993
994 // OR SECOND LOGIC
995

```

```

996 import java.util.*;
997 class FindGCD
998 {
999     int a,b;
1000    void input()
1001    {
1002        Scanner sc=new Scanner(System.in);
1003        System.out.println("ENTER a and b=");
1004        a=sc.nextInt(); //12
1005        b=sc.nextInt(); //8
1006    }
1007    void gcd(FindGCD ob1)
1008    {
1009        int gcd =1;
1010        while(ob1.a!=ob1.b)
1011        for(int i = 1; i <= ob1.a && i <= ob1.b; i++)
1012        {
1013            if(obj.a%i==0 && obj.b%i==0)
1014            {
1015                gcd = i;
1016            }
1017        }
1018        System.out.println("GCD="+gcd);
1019    }
1020 }
1021 class Run
1022 {
1023     public static void main(String args[])
1024     {
1025         FindGCD ob1 = new FindGCD();
1026         ob1.input();
1027
1028         FindGCD ob2 = new FindGCD();
1029         ob2.gcd(ob1);
1030     }
1031 }
1032 =====//END OF CHAP-8//=====
1033 
```