

```

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         }

```

```

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
value,
character value and double value using method overloading */
143 import java.util.Scanner;
144 class Compare
145 {
146     void compare(int a, int b) {
147
148         if (a == b) {
149             System.out.println("value of "+a+" and "+b+" are same");
150         }
151         else {
152             System.out.println("value of "+a+" and "+b+" are not same");
153         }
154     }
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
171 void compare(double a, double b) {
172
173     if (a == b) {
174         System.out.println("value of "+a+" and "+b+" are same");
175     }
176     else {
177         System.out.println("value of "+a+" and "+b+" are not same");
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 }
206 =====array of objects, passing arrays to methods=====
207 //622 Write a Java program to create an array of objects
208 import java.util.*;
209 class ArrayOf {
210     public static void main(String args[]) {
211         //create an array of product object
212         Product[] obj = new Product[5] ;
213         //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
238 array of object).*/
239 class Main{
240     public static void main(String[] arg){
241         Student a[] = new Student[5];
242         for(int i=0; i<5; i++)
243         {
244             a[i] = new Student();
245             System.out.println("----Enter Student "+(i+1)+" Data:----");
246             a[i].getData();
247         }
248         System.out.println("=====");
249         System.out.println("===Student Details=====");
250         System.out.println("=====");
251         for(int i=0; i<5; i++)
252         {
253             a[i].printData();
254         }
255     }
256 }
257 class Student
258 {
259     int RN;
260     String name;
261     long Mo_no;
262     void getData()
263     {
264         Scanner sc = new Scanner(System.in);
265         System.out.print("Enter Roll no: ");
266         RN = sc.nextInt();
267         sc.nextLine();
268         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
arrays to methods.
386
387 class arrayDim{
388     void display(int[] array){
389         System.out.println("Yor array is:");
390         for(int q: array)
391             System.out.print(q);
392     }
393     void display(int[][] array){
394         System.out.println("Yor array is:");
395         for(int j[]: array)
396             for(int i: j)
397                 System.out.print(i);
398     }
399 }
400 class Run{
401     public static void main(String[] args) {
402         int[] a = {11,12,13};
403         int[][] b = {{1,2,3},{1,2,3},{1,2,3}};
404         arrayDim ad = new arrayDim();
405         ad.display(a);
406         ad.display(b);
407     }
408 }
409
410 //627 Write a Java program to find Even number and Odd number from given Array using the
concept of passing arrays to methods
411 class OddEven{
412     public static void main(String args[]){
413         int arr[]={1,2,5,6,3,2};
414         OddEven oe = new OddEven();
415         oe.oddprint(arr);
416         oe.evenprint(arr);
417     }
418     void oddprint(int a[]){
419         System.out.println("Odd Numbers:");
420         for(int i=0;i<a.length;i++){
421             if(a[i]%2!=0){
422                 System.out.println(a[i]);
423             }
424         }
425     }
426 }

```



```

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
arrays to methods.
438 import java.util.Scanner;
439 class Array {
440     public int max(int[] array) {
441         int max = array[0];
442
443         for (int i = 0; i < array.length; i++) {
444             if (array[i] > max) {
445                 max = array[i];
446             }
447         }
448         return max;
449     }
450     public int min(int[] array) {
451         int min = array[0];
452         for (int i = 0; i < array.length; i++) {
453             if (array[i] < min) {
454                 min = array[i];
455             }
456         }
457         return min;
458     }
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
elements using class. Here, use concept of passing array as argument to method. Use
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
563 /*637 Write a Java program to add 5 in original value of array elements and display
564 modified elements of
565 one dimensional array using pass by value mechanism. */
566 class Array {
567     public static void main(String[] args) {
568         int a[] = {10, 20, 30, 40, 50};
569         Array obj = new Array();
570         obj.modifyArray(a);
571         for (int i = 0; i < a.length; i++) {
572             obj.print(a[i]);
573         }
574     }
575     void modifyArray(int a[]) {
576         for (int i = 0; i < a.length; i++) {
577             a[i] += 5;
578         }
579     }
580     void print(int a) {
581         System.out.println(a);
582     }
583 }
584
585 /*638 Write a Java program to change original value, which is given from user by using
586 appropriate Parameter passing technique. */
587 class CallByReference {
588     int a, b;
589     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
value
607 import java.util.Scanner;
608 class SwapNumber {
609     public static void main(String args[]) {
610         int x, y;
611         Scanner sc = new Scanner(System.in);
612         System.out.print("Enter the first number: ");
613         x = sc.nextInt();
614         System.out.print("Enter the second number: ");
615         y = sc.nextInt();
616         System.out.println("Before Calling in main\nx = "+x+"\ny = "+y);
617         SwapNumber s = new SwapNumber();
618         s.swap(x, y);
619         System.out.println("After calling in main\nx = "+x+"\ny = "+y); // no effect of
swap occurs in x,y here
620     }
621     void swap(int a, int b) {
622         a = a + b;
623         b = a - b;
624         a = a - b;
625         System.out.println("After Swapping inside method\nx = "+a+"\ny = "+b); // swap
occurs here only in a,b but not in original values x,y
626     }
627 }
628
629 //640 Write a Java program to Swap two values, which is given from user by using call by
reference.
630 class CallByReference {
631     int a, b;
632     void set(int x, int y) {
633         a = x;
634         b = y;
635     }
636     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: " + obj.b); //swap occurs in a,b of obj as well as in a,b of cbr
642     }
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: " + cbr.b); // values set in instance of cbr.
650         cbr.swapvalue(cbr);
651         System.out.println("After calling in main\n Value of a: " + cbr.a+ " & b: " + cbr.b); // effect of obj swapping occurs in cbr also here
652     }
653 }
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
reference.
704 class Array{
705     public static void main(String[] args) {
706         int a[] = {10,20,30,40,50};
707         int b[] = a;
708         Array obj = new Array();
709         obj.print(b);
710     }
711     void print(int a[]){
712         for (int i = 0; i<a.length;i++){
713             System.out.println(a[i]);
714         }
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
and meters) and display the result in a user-readable format.
722 2. Classes and Methods:
723     - `Distance` class:
724         - `int km, m`: Instance variables to store kilometers and meters.
725         - `Scanner sc`: Used to take input from the user.
726         - `void get()`: Method to input distance values from the user.
727         - `void show(int a, int b)`: Method to display the distance in kilometers and
meters.
728         - `Distance add(Distance obj1, Distance obj2)`: Method to add two distances and
return the result as a new `Distance` object.
729     - `run` class:
730         - `main(String args[])`: The main method where the program execution begins.
731         - Creates instances of the `Distance` class, takes input for two distances,
calculates their sum, and displays the result.
732 3. Input and Output:
733     - The program prompts the user to input two distances: one in kilometers and another
in meters (or 0 if no meters are present).
734     - After input, it calculates the sum and displays it in kilometers and meters.
735 4. Distance Calculation:
736     - 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.*/*

```
import java.util.*;
class Distance
{
    int km,m;
    Scanner sc = new Scanner(System.in);
    void get()
    {
        System.out.println("Enter value of distance in KM:");
        km= sc.nextInt();

        System.out.println("Enter value of distance in meter if any otherwise enter 0:");
        m = sc.nextInt();
    }

    void show(int a,int b)
    {
        System.out.println("Distance = "+a+"."+b+"KM");
    }

    Distance add(Distance obj1,Distance obj2)
    {
        Distance d4 = new Distance();
        d4.km = obj1.km + obj2.km;
        d4.m = obj1.m + obj2.m;

        if(d4.m>=1000)
        {
            d4.km+=1;
            d4.m-=1000;
        }
        return d4;
    }
}

class run
{
    public static void main(String args[])
    {
        Distance d1 = new Distance();
        Distance d2 = new Distance();
        Distance d3 = new Distance();
        Distance d4 = new Distance();
        d1.get();

        d1.show(d1.km,d1.m);

        d2.get();
        d2.show(d2.km,d2.m);
    }
}
```

```

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

```



```

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
method(i.e num=5 and power=3 then ans is5^3 that is 125)
924
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
941 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
method(i.e a=4 and b=6 then GCD is 2)
959 //FIRST LOGIC
960 import java.util.*;
961 class FindGCD
962 {
963     int a,b;
964     void input()
965     {
966         Scanner sc=new Scanner(System.in);
967         System.out.println("ENTER a and b=");
968         a=sc.nextInt();
969         b=sc.nextInt();
970     }
971     void gcd(FindGCD ob1)
972     {
973         while(ob1.a!=ob1.b)
974         {
975             if(ob1.a>=ob1.b) //when first num is bigger, assign in first
976                 ob1.a=ob1.a-ob1.b;
977             else //when second num is bigger, assign in second
978                 ob1.b=ob1.b-ob1.a;
979         }
980         System.out.println("GCD="+ob1.a);
981     }
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 obl)
1008     {
1009         int gcd =1;
1010         while(obl.a!=obl.b)
1011             for(int i = 1; i <= obl.a && i <= obl.b; i++)
1012             {
1013                 if(obl.a%i==0 && obl.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 obl = new FindGCD();
1026         obl.input();
1027
1028         FindGCD ob2 = new FindGCD();
1029         ob2.gcd(obl);
1030     }
1031 }
1032
1033 =====//END OF CHAP-8//=====

```