

VISVESVARAYA TECHNOLOGICAL UNIVERSITY
“JnanaSangama”, Belgaum -590014, Karnataka.



LAB REPORT
on
Object Oriented Java Programming
(23CS3PCOOJ)

Submitted by

Anmol Bhattarai (**1BM23CS039**)

in partial fulfillment for the award of the degree of
BACHELOR OF ENGINEERING
in
COMPUTER SCIENCE AND ENGINEERING



B.M.S. COLLEGE OF ENGINEERING
(Autonomous Institution under VTU)
BENGALURU-560019

Sep-2024 to Jan-2025

**B.M.S. College of Engineering,
Bull Temple Road, Bangalore 560019**
(Affiliated To Visvesvaraya Technological University, Belgaum)
Department of Computer Science and Engineering



CERTIFICATE

This is to certify that the Lab work entitled “Object Oriented Java Programming (23CS3PCOOJ)” carried out by **Anmol Bhattacharai (1BM23CS039)**, who is a bonafide student of **B.M.S. College of Engineering**. It is in partial fulfillment for the award of **Bachelor of Engineering in Computer Science and Engineering** of the Visvesvaraya Technological University, Belgaum. The Lab report has been approved as it satisfies the academic requirements in respect of an Object Oriented Java Programming (23CS3PCOOJ) work prescribed for the said degree.

Dr. Prasad Gr Assistant Professor Department of CSE, BMSCE	Dr. Jyothi S Nayak Professor & HOD Department of CSE, BMSCE
--	---

Index

Sl. No.	Date	Experiment Title	Page No.
1	9/10/24	Implement Quadratic Equation	4
2	16/10/24	Implement SGPA Calculator	9
3	23/10/24	Create Objects for Books	16
4	30/10/24	Implement Abstract Class	23
5	6/11/24	Bank Account Management	28
6	13/11/24	Implement Packages	40
7	20/11/24	Implement Exception Handling	51
8	27/11/24	Multithreading, Creating Threads in Java	57
9	27/11/24	Interface to Perform Integer Division	60
10	27/11/24	Implement Deadlock Implement Inter-process Communication	65

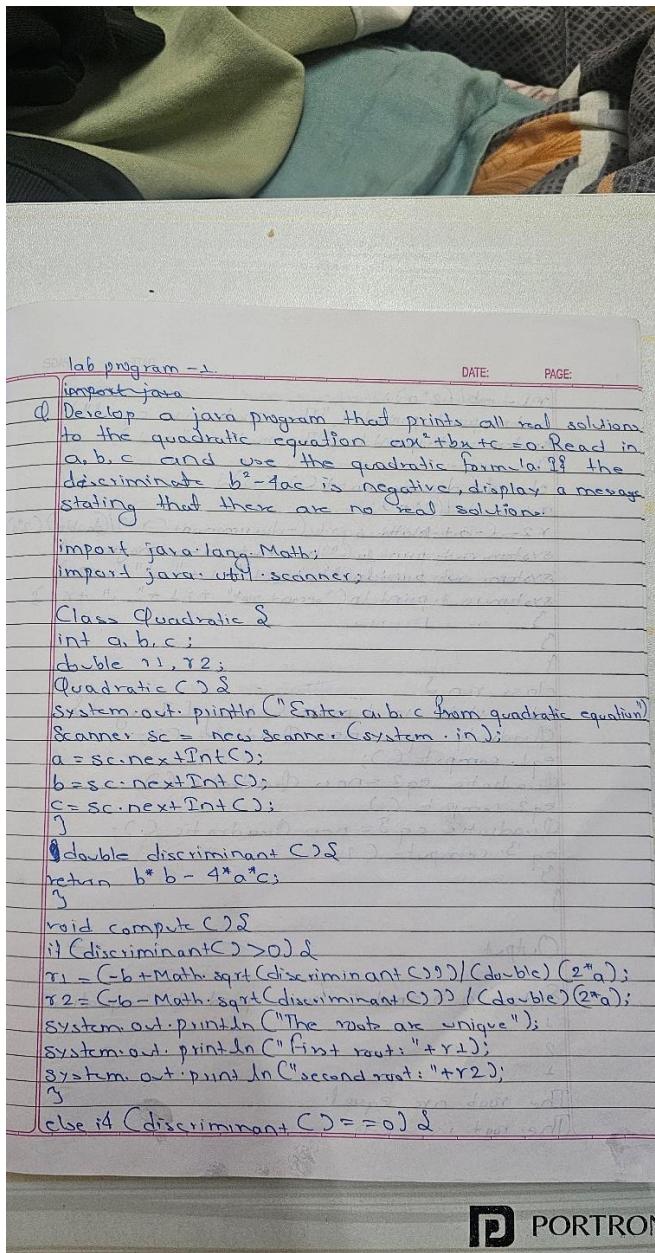
Github Link:

<https://github.com/Anmolbhattara/JAVA>

Program 1

Implement Quadratic Equation

Algorithm:



```
DATE: PAGE:  
r1 = -b / (2 * a);  
System.out.println("The root are equal");  
System.out.println("The root is: " + r1);  
}  
else if (discriminant() < 0) {  
    r1 = -b / (2 * a);  
    r2 = (-b + Math.sqrt(-discriminant())) / (2 * a);  
    System.out.println("The root are imaginary");  
    System.out.println("First root: " + r1 + " + " + r2);  
    System.out.println("Second root: " + r1 + " - " + r2);  
}  
}  
}  
}  
}  
}  
class run {  
    public static void main (String [] args) {  
        Quadratic cq1 = new Quadratic ();  
        cq1.compute ();  
        Quadratic cq2 = new Quadratic ();  
        cq2.compute ();  
        Quadratic cq3 = new Quadratic ();  
        cq3.compute ();  
    }  
}  
  
Output:  
Enter a,b and c from quadratic equation:  
1  
2  
3  
The root are equal.  
The root is: -1.0
```


Code:

```
import java.util.Scanner;

public class Main {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        System.out.print("Enter coefficient a: ");
        double a = scanner.nextDouble();

        System.out.print("Enter coefficient b: ");
        double b = scanner.nextDouble();

        System.out.print("Enter coefficient c: ");
        double c = scanner.nextDouble();

        double discriminant = b * b - 4 * a * c;

        if (discriminant > 0) {
            double root1 = (-b + Math.sqrt(discriminant)) / (2 * a);
            double root2 = (-b - Math.sqrt(discriminant)) / (2 * a);
            System.out.println("The roots are real and different:");
            System.out.println("Root 1: " + root1);
            System.out.println("Root 2: " + root2);
        } else if (discriminant == 0) {
            double root = -b / (2 * a);
            System.out.println("The roots are real and the same:");
            System.out.println("Root: " + root);
        } else {
            System.out.println("The roots are complex:");
            double realPart = -b / (2 * a);
            double imaginaryPart = Math.sqrt(-discriminant) / (2 * a);
            System.out.println("Root 1: " + realPart + " + " + imaginaryPart + "i");
            System.out.println("Root 2: " + realPart + " - " + imaginaryPart + "i");
            System.out.print("Ayush Ranjan 1BM23CS058");
        }
    }

    scanner.close();
}
```

Output :

Enter a, b, c:

1

2

1

Equal roots. $r1 = r2 = -1.0$

Name: Anmol Bhattacharai USN: 1BM23CS039

Enter a, b, c:

1

-7

12

$r1 = 6.0$ $r2 = 2.0$

Name: Anmol Bhattacharai USN: 1BM23CS039

Enter a, b, c:

1

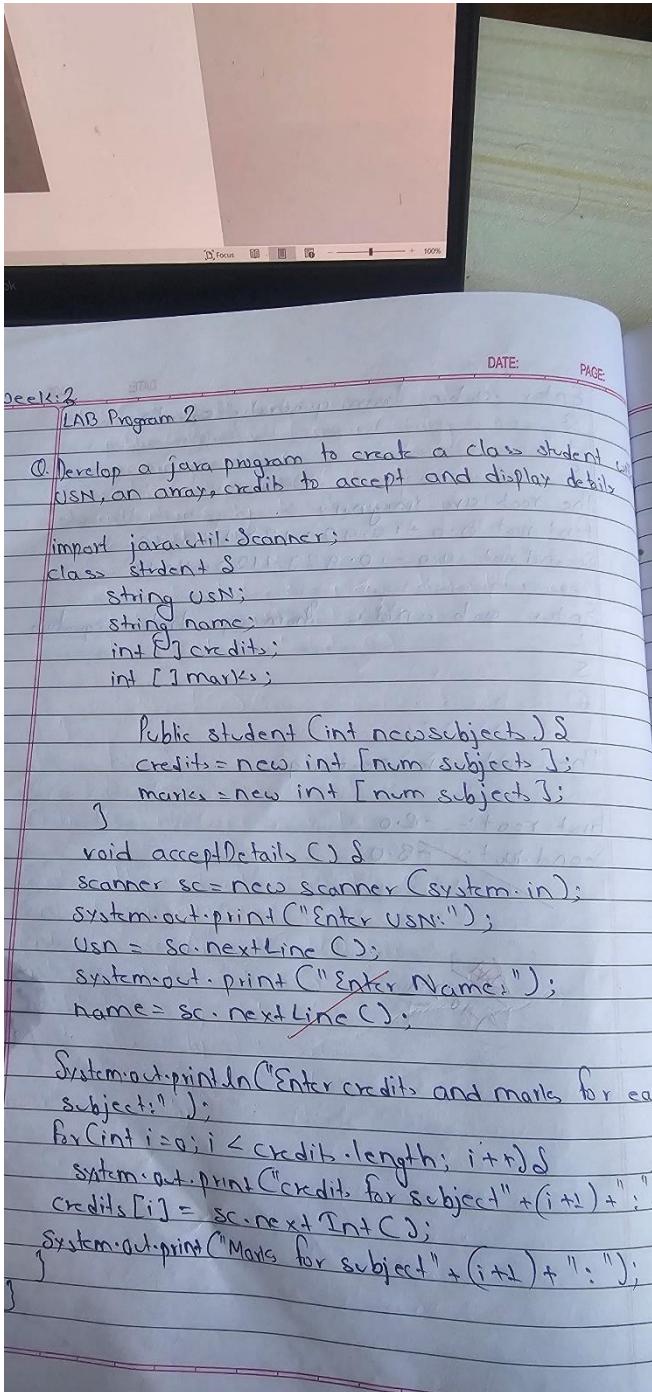
2

5

Roots are imaginary

Name: Anmol Bhattacharai USN: 1BM23CS039

Program 2 :



Implement SGPA Calculator

Algorithm :

DATE: PAGE:

```

Public class StudentSGPAcalculator {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter the number of subjects: ");
        int numSubjects = scanner.nextInt();
        scanner.nextLine();
        Student student = new Student(numSubjects);
        student.acceptDetails();
        student.displayDetails();
        student.close();
    }
}

class Student {
    int numSubjects;
    String usn;
    String name;
    double credits;
    double marks;
    double totalCredits;
    double totalMarks;
    double sgpa;

    public void acceptDetails() {
        System.out.print("Enter USN: ");
        usn = System.console().readLine();
        System.out.print("Enter Name: ");
        name = System.console().readLine();
        System.out.print("Enter credits and marks for each subject: ");
        for (int i = 0; i < numSubjects; i++) {
            System.out.print("Credit for subject " + (i + 1) + ": ");
            credits = Double.parseDouble(System.console().readLine());
            System.out.print("Marks for subject " + (i + 1) + ": ");
            marks = Double.parseDouble(System.console().readLine());
            totalCredits += credits;
            totalMarks += marks;
        }
    }

    public void displayDetails() {
        System.out.println("USN: " + usn);
        System.out.println("Name: " + name);
        System.out.println("Total Credits: " + totalCredits);
        System.out.println("Total Marks: " + totalMarks);
        System.out.println("SGPA: " + sgpa);
    }

    public void close() {
        System.out.println("Closing the student object.");
    }
}

```

Output:

```

Enter the no. of subjects: 2
Enter USN: 1BM23C079
Enter name: Anmol Bhattarai
Enter credit and marks for each subject:
Credit for subject 1: 3
marks for subject 1: 90
Credit for subject 2: 4
marks for subject 2: 90
SGPA: 9

```

DATE: PAGE:

```

void displayDetails() {
    System.out.println("USN: " + usn);
    System.out.println("Name: " + name);
    for (int i = 0; i < credit.length; i++) {
        System.out.print("subject " + i + " Marks: " + marks[i] + " - " +
                        "credits: " + credits[i] + " Marks: " + marks[i]);
    }
    System.out.print("SGPA: " + calculateSGPA());
}

double calculateSGPA() {
    double totalCredits = 0;
    double totalMarks = 0;
    for (int i = 0; i < credit.length; i++) {
        double gradePoint = calculateGradePoint(marks[i]);
        totalPoint += gradePoint * credit[i];
        totalCredit += credit[i];
    }
    return totalPoint / totalCredit;
}

double calculateGradePoint(int mark) {
    if (mark >= 90) return 10;
    else if (mark >= 80) return 9;
    else if (mark >= 70) return 8;
    else if (mark >= 60) return 7;
    else if (mark >= 50) return 6;
    else if (mark >= 40) return 5;
    else return 0;
}

```

Code :

```
import java.util.*;

class Stud_details {
    int marks[] = new int[8];
    int credit[] = new int[8];
    String usn, name;
    Scanner sc = new Scanner(System.in);

    void getdetails() {
        System.out.println("Enter the usn and name:");
        usn = sc.next();
        name = sc.next();
        System.out.println("Enter the marks for 8 subjects:");
        for (int i = 0; i < 8; i++) {
            marks[i] = sc.nextInt();
        }
        System.out.println("Enter the credits for 8 subjects:");
        for (int i = 0; i < 8; i++) {
            credit[i] = sc.nextInt();
        }
    }

    void display() {
        System.out.println("\nStudent Details:");
        System.out.println("USN: " + usn);
        System.out.println("Name: " + name);
        for (int i = 0; i < 8; i++) {
            System.out.println("Marks of Subject " + (i + 1) + ": " + marks[i]);
        }
        System.out.println("SGPA: " + calculateSGPA());
    }

    double getgradepoint(int mark) {
        if (mark >= 90) return 10.0;
        else if (mark >= 80) return 9.0;
        else if (mark >= 70) return 8.0;
    }
}
```

```

        else if (mark >= 60) return 7.0;
        else if (mark >= 50) return 6.0;
        else if (mark >= 40) return 5.0;
        else return 0.0;
    }

    double calculateSGPA() {
        int totalcredits = 0;
        double gradepoint = 0;
        for (int i = 0; i < 8; i++) {
            totalcredits += credit[i];
        }
        for (int i = 0; i < 8; i++) {
            gradepoint += getgradepoint(marks[i]) * credit[i];
        }
        return (gradepoint / totalcredits);
    }
}

public class Student {
    public static void main(String args[]) {
        Stud_details s1[] = new Stud_details[3];
        for (int j = 0; j < 3; j++) {
            s1[j] = new Stud_details();
        }
        for (int j = 0; j < 3; j++) {
            System.out.println("\nEnter details of student " + (j + 1));
            s1[j].getdetails();
        }
        for (int j = 0; j < 3; j++) {
            s1[j].display();
        }
        System.out.println("\nName: Anmol Bhattarai USN: 1BM23CS058");
    }
}

```

Output :

Enter details of student 1

Enter the usn and name:

1BM23CS001

John

Enter the marks for 8 subjects:

90

80

70

60

50

40

30

20

Enter the credits for 8 subjects:

4

3

3

3

2

3

2

3

3

Enter details of student 2

Enter the usn and name:

1BM23CS002

Alice

Enter the marks for 8 subjects:

80

85

75

65

55

45

35

25

Enter the credits for 8 subjects:

4
3
3
3
2
3
2
3

Enter details of student 3

Enter the usn and name:

1BM23CS003

Bob

Enter the marks for 8 subjects:

85
90
95
80
70
60
50
40

Enter the credits for 8 subjects:

4
3
3
3
2
3
2
3

Student Details:

USN: 1BM23CS001

Name: John

Marks of Subject 1: 90

Marks of Subject 2: 80

Marks of Subject 3: 70

Marks of Subject 4: 60

Marks of Subject 5: 50

Marks of Subject 6: 40

Marks of Subject 7: 30

Marks of Subject 8: 20

SGPA: 5.55555555555555

Student Details:

USN: 1BM23CS002

Name: Alice

Marks of Subject 1: 80

Marks of Subject 2: 85

Marks of Subject 3: 75

Marks of Subject 4: 65

Marks of Subject 5: 55

Marks of Subject 6: 45

Marks of Subject 7: 35

Marks of Subject 8: 25

SGPA: 5.5

Student Details:

USN: 1BM23CS003

Name: Bob

Marks of Subject 1: 85

Marks of Subject 2: 90

Marks of Subject 3: 95

Marks of Subject 4: 80

Marks of Subject 5: 70

Marks of Subject 6: 60

Marks of Subject 7: 50

Marks of Subject 8: 40

SGPA: 7.722222222222222

Name: Anmol Bhattacharai USN: 1BM23CS039

Program 3 :
Create Objects for Books

Algorithm:

Lab: 3. JTAO

DATE: PAGE:

Q Create a class `book` which contains four members: `name`, `author`, `price`, `numPages`. Include a constructor to set the values for the members. Include methods to set and get the details of the objects. Include `toString()` method that could display the complete details of the book. Develop a Java program to create book objects.

Explore `toString()` method usage in Java.

```
import java.util.Scanner; import java.util.*;  
class book {  
    String name, author; int numPages;  
    double price;  
    void Book (String name, String author, int numPages, double price) {  
        this.name = name;  
        this.author = author;  
        this.numPages = numPages;  
        this.price = price;  
    }  
    void setDetails() {  
        Scanner sc = new Scanner (System.in);  
        System.out.print ("Enter name:");  
        this.name = sc.next();  
        System.out.print ("Enter author:");  
        this.author = sc.next();  
        System.out.print ("Enter pages:");  
        this.numPages = sc.nextInt();  
        System.out.print ("Enter price:");  
        this.price = sc.nextDouble();  
    }  
    void getDetails() {  
        System.out.println ("Name: " + name + " Author: " + author + " numPages: " + numPages + " price: " + price);  
    }  
}
```

DATE: 8/10/2018 PAGE: 5

```

Public String toString() {
    return "Name: " + name + " Author: " + author + " In page: "
           + numPages + " In price: " + price;
}

class BookDemo {
    public static void main(String args[]) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter the number of books: ");
        int bookNum = sc.nextInt();
        Book bookArray[] = new Book[bookNum];
        for (int i = 0; i < bookNum; i++) {
            bookArray[i] = new Book();
            bookArray[i].setDetails();
            System.out.println();
        }
        for (int i = 0; i < bookNum; i++) {
            bookArray[i].getDetails();
        }
        System.out.println("Total price of books is: " + bookArray[0].getPrice());
    }
}

class Book {
    String name;
    String author;
    int numPages;
    int price;

    public void setDetails() {
        System.out.print("Enter book name: ");
        name = sc.nextLine();
        System.out.print("Enter book author: ");
        author = sc.nextLine();
        System.out.print("Enter book numPages: ");
        numPages = sc.nextInt();
        System.out.print("Enter book price: ");
        price = sc.nextInt();
    }

    public String toString() {
        return name + " " + author + " " + numPages + " " + price;
    }
}

```

DATE:

Output: Enter the name of book

Enter the number of books: 2

Enter name: Pratik

Enter author: aniket

Enter pages: 200

Enter price: 180

Enter name: Sahil

Enter Author: Sunil

Enter Page: 210

Enter Price: 300

Name: Pratik

Author: aniket

Page: 200

Price: 180

Name: Sahil

Author: sunil

Page: 210

Price: 300

Code :

```
import java.util.*;  
  
class Book {  
    String name;  
    String author;  
    int price;  
    int num_pages;  
  
    Book(String name, String author, int price, int num_pages) {  
        this.name = name;  
        this.author = author;  
        this.price = price;  
        this.num_pages = num_pages;  
    }  
  
    public String toString() {  
        return "Name: " + name + ", Author: " + author + ", Price: " + price + ",  
        No of Pages: " + num_pages;  
    }  
}  
  
class BookDetails {  
    public static void main(String[] args) {  
        Scanner sc = new Scanner(System.in);  
  
        System.out.println("Enter number of books:");  
        int n = sc.nextInt();  
        sc.nextLine();  
  
        Book[] book = new Book[n];  
  
        System.out.println("Enter name, author, price and number of pages for  
        each book respectively:");  
        for (int i = 0; i < n; i++) {  
            System.out.println("Enter details of book " + (i + 1) + ":");  
  
            System.out.print("Name: ");  
            String name = sc.nextLine();
```

```
System.out.print("Author: ");
String author = sc.nextLine();

System.out.print("Price: ");
int price = sc.nextInt();

System.out.print("Number of Pages: ");
int num_pages = sc.nextInt();
sc.nextLine();

book[i] = new Book(name, author, price, num_pages);
}

for (int i = 0; i < n; i++) {
    System.out.println("\nBook " + (i + 1) + ":");
    System.out.println(book[i].toString());
}

sc.close();
}
```

o/p

Enter number of books:

2

Enter name, author, price and number of pages for each book respectively:

Enter details of book 1:

Name: Java Programming

Author: James Gosling

Price: 500

Number of Pages: 300

Enter details of book 2:

Name: Data Structures

Author: Robert Lafore

Price: 400

Number of Pages: 450

Book 1:

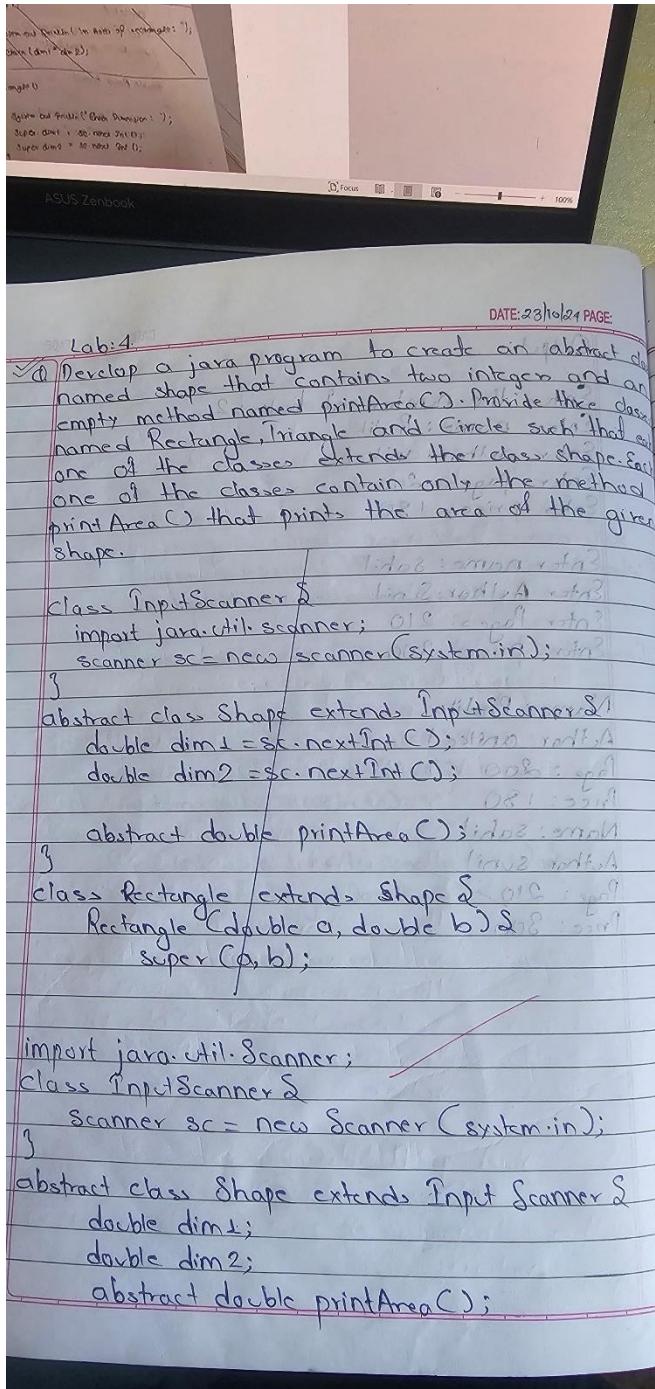
Name: Java Programming, Author: James Gosling, Price: 500, No of Pages: 300

Book 2:

Name: Data Structures, Author: Robert Lafore, Price: 400, No of Pages: 450

Program 4 : Implement Abstract Class

Algorithm:



303 300 DATE PAGE

```

} 3.0000000000000004
class Rectangle extends Shape {
    Rectangle() {
        super();
        System.out.println("Enter the dimension of rectangle");
        super.dim1 = sc.nextInt();
        super.dim2 = sc.nextInt();
    }
    double printArea() {
        System.out.println("Area of rectangle:");
        return (dim1 * dim2);
    }
}

class Triangle extends Shape {
    Triangle() {
        System.out.println("Enter the dimension of triangle");
        super.dim1 = sc.nextInt();
        super.dim2 = sc.nextInt();
    }
    double printArea() {
        System.out.println("Area of triangle:");
        return 0.5 * dim1 * dim2;
    }
}

class Circle extends Shape {
    Circle() {
        System.out.println("Enter the dimension of the circle");
        super.dim1 = sc.nextInt();
    }
    double printArea() {
        System.out.println("Area of circle:");
        return 3.14 * dim1 * dim1;
    }
}

```

no and possible ("Brick Dimensions")
for count = 50, need 701.0;
for count = 50, need 701.1;

Focus . 100%

ASUS Zenbook

DATE: PAGE:

class AbstractDemo 8
public static void main (String args []) {
 Rectangle r = new Rectangle (10, 20);
 Triangle t = new Triangle (5);
 Circle c = new Circle (5);
 Shape figure;
 System.out.println ("Area is: " + figure.printArea());
 figure = t;
 System.out.println ("Area is: " + figure.printArea());
 figure = c; // error: cast from Triangle to Circle
 System.out.println ("Area is: " + figure.printArea());
}
} // error: no main method
Enter the dimension of the Rectangle:
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
779
780
781
782
783
784
785
786
787
788
789
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
809
810
811
812
813
814
815
816
817
818
819
819
820
821
822
823
824
825
826
827
828
829
829
830
831
832
833
834
835
836
837
838
839
839
840
841
842
843
844
845
846
847
848
849
849
850
851
852
853
854
855
856
857
858
859
859
860
861
862
863
864
865
866
867
868
869
869
870
871
872
873
874
875
876
877
878
879
879
880
881
882
883
884
885
886
887
888
889
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
909
910
911
912
913
914
915
916
917
918
919
919
920
921
922
923
924
925
926
927
928
929
929
930
931
932
933
934
935
936
937
938
939
939
940
941
942
943
944
945
946
947
948
949
949
950
951
952
953
954
955
956
957
958
959
959
960
961
962
963
964
965
966
967
968
969
969
970
971
972
973
974
975
976
977
978
979
979
980
981
982
983
984
985
986
987
988
989
989
990
991
992
993
994
995
996
997
998
999
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1059
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1079
1079
1080
1081
1082
1083
1084
1085
1086
1087
1088
1089
1089
1090
1091
1092
1093
1094
1095
1096
1097
1098
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1109
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1129
1130
1131
1132
1133
1134
1135
1136
1137
1138
1139
1139
1140
1141
1142
1143
1144
1145
1146
1147
1148
1149
1149
1150
1151
1152
1153
1154
1155
1156
1157
1158
1159
1159
1160
1161
1162
1163
1164
1165
1166
1167
1168
1169
1169
1170
1171
1172
1173
1174
1175
1176
1177
1178
1179
1179
1180
1181
1182
1183
1184
1185
1186
1187
1188
1189
1189
1190
1191
1192
1193
1194
1195
1196
1197
1198
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1249
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
1259
1260
1261
1262
1263
1264
1265
1266
1267
1268
1269
1269
1270
1271
1272
1273
1274
1275
1276
1277
1278
1279
1279
1280
1281
1282
1283
1284
1285
1286
1287
1288
1289
1289
1290
1291
1292
1293
1294
1295
1296
1297
1298
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1309
1310
1311
1312
1313
1314
1315
1316
1317
1318
1319
1319
1320
1321
1322
1323
1324
1325
1326
1327
1328
1329
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1339
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1349
1349
1350
1351
1352
1353
1354
1355
1356
1357
1358
1359
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1369
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378
1379
1379
1380
1381
1382
1383
1384
1385
1386
1387
1388
1389
1389
1390
1391
1392
1393
1394
1395
1396
1397
1398
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1419
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1429
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1439
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1449
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1459
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1469
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1479
1479
1480
1481
1482
1483
1484
1485
1486
1487
1488
1489
1489
1490
1491
1492
1493
1494
1495
1496
1497
1498
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
1509
1510
1511
1512
1513
1514
1515
1516
1517
1518
1519
1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1529
1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1539
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1549
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559
1559
1560
1561
1562
1563
1564
1565
1566
1567
1568
1569
1569
1570
1571
1572
1573
1574
1575
1576
1577
1578
1579
1579
1580
1581
1582
1583
1584
1585
1586
1587
1588
1589
1589
1590
1591
1592
1593
1594
1595
1596
1597
1598
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1609
1609
1610
1611
1612
1613
1614
1615
1616
1617
1618
1619
1619
1620
1621
1622
1623
1624
1625
1626
1627
1628
1629
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1639
1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1649
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1659
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1669
1669
1670
1671
1672
1673
1674
1675
1676
1677
1678
1679
1679
1680
1681
1682
1683
1684
1685
1686
1687
1688
1689
1689
1690
1691
1692
1693
1694
1695
1696
1697
1698
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708
1709
1709
1710
1711
1712
1713
1714
1715
1716
1717
1718
1719
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1729
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738
1739
1739
1740
1741
1742
1743
1744
1745
1746
1747
1748
1749
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1769
1769
1770
1771
1772
1773
1774
1775
1776
1777
1778
1779
1779
1780
1781
1782
1783
1784
1785
1786
1787
1788
1789
1789
1790
1791
1792
1793
1794
1795
1796
1797
1798
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1809
1810
1811
1812
1813
1814
1815
1816
1817
1818
1819
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828
1829
1829
1830
1831
1832
1833
1834
1835
1836
1837
1838
1839
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1849
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858
1859
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1869
1869
1870
1871
1872
1873
1874
1875
1876
1877
1878
1879
1879
1880
1881
1882
1883
1884
1885
1886
1887
1888
1889
1889
1890
1891
1892
1893
1894
1895
1896
1897
1898
1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
1909
1910
1911
1912
1913
1914
1915
1916
1917
1918
1919
1919
1920
1921
1922
1923
1924
1925
1926
1927
1928
1929
1929
1930
1931
1932
1933
1934
1935
1936
1937
1938
1939
1939
1940
1941
1942
1943
1944
1945
1946
1947
1948
1949
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
1979
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2019
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
2029
2030
2031
2032
2033
2034
2035
2036
2037
2038
2039
2039
2040
2041
2042
2043
2044
2045
2046
2047
2048
2049
2049
2050
2051
2052
2053
2054
2055
2056
2057
2058
2059
2059
2060
2061
2062
2063
2064
2065
2066
2067
2068
2069
2069
2070
2071
2072
2073
2074
2075
2076
2077
2078
2079
2079
2080
2081
2082
2083
2084
2085
2086
2087
2088
2089
2089
2090
2091
2092
2093
2094
2095
2096
2097
2098
209

DATE: _____

Area of rectangle = $3 \times 2 = 6$ sq. cm

Area is 12 cm^2 with dimensions 3 cm by 4 cm

Area of Triangle = $\frac{1}{2} \times 3 \times 4 = 6$ sq. cm

Area is 30 cm^2 with 6 cm and 10 cm

Area of circle = πr^2 with 3 cm radius

Area is 28.26 cm^2

~~Area of square = $2 \times 2 = 4$ sq. cm~~

~~Area of rectangle = $3 \times 2 = 6$ sq. cm~~

~~Area of triangle = $\frac{1}{2} \times 3 \times 4 = 6$ sq. cm~~

~~Area of circle = πr^2 with 3 cm radius~~

~~Area is 28.26 cm^2~~

Code :

```
import java.util.*;  
  
abstract class Shape {  
    int dim1, dim2;  
  
    Shape(int a, int b) {  
        dim1 = a;  
        dim2 = b;  
    }  
  
    abstract void printArea();  
}  
  
class Rectangle extends Shape {  
    Rectangle(int l, int b) {  
        super(l, b);  
    }  
  
    void printArea() {  
        System.out.println("Area of Rectangle: " + (double)(dim1 * dim2));  
    }  
}  
  
class Triangle extends Shape {  
    Triangle(int b, int h) {  
        super(b, h);  
    }  
  
    void printArea() {  
        System.out.println("Area of Triangle: " + (double)(dim1 * dim2 / 2));  
    }  
}  
  
class Circle extends Shape {  
    Circle(int r) {  
        super(r, 0);  
    }  
  
    void printArea() {  
        System.out.println("Area of Circle: " + (Math.PI * dim1 * dim1));  
    }  
}
```

```

class ShapeArea {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        System.out.println("Enter length and width of rectangle:");
        int length = sc.nextInt();
        int breadth = sc.nextInt();

        System.out.println("Enter base and height of triangle:");
        int base = sc.nextInt();
        int height = sc.nextInt();

        System.out.println("Enter radius of circle:");
        int radius = sc.nextInt();

        Rectangle r = new Rectangle(length, breadth);
        Triangle t = new Triangle(base, height);
        Circle c = new Circle(radius);

        r.printArea();
        t.printArea();
        c.printArea();

        sc.close();
    }
}

```

o/p

Enter length and width of rectangle:
10 5

Enter base and height of triangle:
8 6

Enter radius of circle:
7

Area of Rectangle: 50.0
Area of Triangle: 24.0
Area of Circle: 153.93804002589985

Program 5 : Bank Account Management

Algorithm :

Lab: 5 DATE: 3/10/2022 PAGE:

```
# Bank Account Management
import java.util.Scanner;

class Account {
    String customerName, accountType;
    int accountNumber;
    double balance;

    Account(String customerName, int accountNumber, String accountType, double initialBalance) {
        this.customerName = customerName;
        this.accountNumber = accountNumber;
        this.accountType = accountType;
        this.initialBalance = initialBalance;
    }

    void deposit(double amount) {
        balance += amount;
        System.out.println("Deposit successful. New balance: $" + balance);
    }

    void displayBalance() {
        System.out.println("Current balance: $" + balance);
    }
}

class SavingsAccount extends Account {
    private static final double InterestRate = 0.04;
    SavingsAccount(String customerName, int accountNumber, double initialBalance) {
        super(customerName, accountNumber, "Savings", initialBalance);
    }
}
```

DATE: _____ PAGE: _____

```

Super(customerName, accountNumber, accountType,
      "Savings", initialBalance);

void computeInterest() {
    double interest = balance * interestRate;
    balance += interest;
    System.out.println("Interest added. New balance: $" +
                       balance);
}

void withdraw(double amount) {
    if (amount > balance) {
        System.out.println("Insufficient balance.");
    } else {
        balance -= amount;
        System.out.println("Withdrawal successful. New balance:
                           $" + balance);
    }
}

class CurrentAccount extends Account {
    private static final double minBalance = 500.0;
    private static final double serviceCharge = 2.5;
}

CurrentAccount(String customerName, int accountNumber,
               double initialBalance) {
    Super(customerName, accountNumber, accountType,
          initialBalance);
}

```

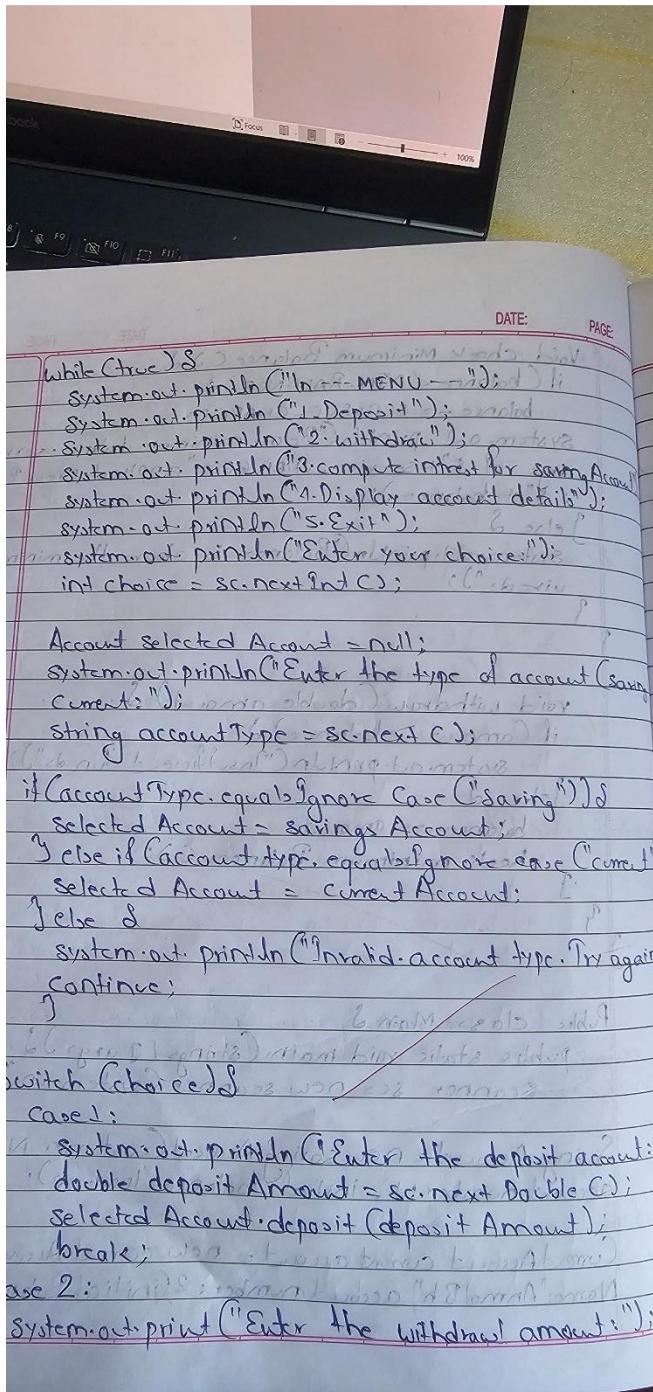
```

DATE: PAGE:
Void check Minimum Balance () {
    if (balance < MIN_BALANCE) {
        balance -= SERVICE_CHARGE;
        System.out.println("Below minimum balance. service charge of " + SERVICE_CHARGE + " applied. New Balance is " + balance);
    } else {
        System.out.println("Balance is above the minimum required.");
    }
}

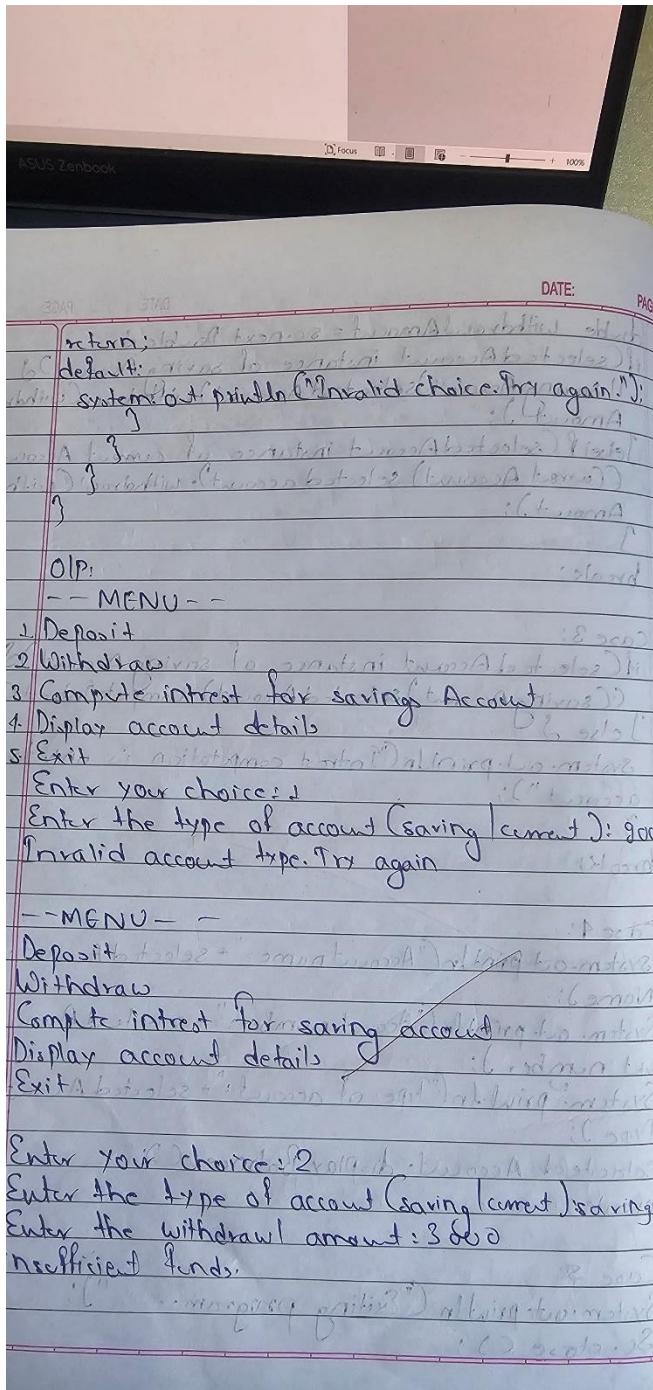
void withdraw (double amount) {
    if (amount > balance) {
        System.out.println("Insufficient funds");
    } else {
        balance -= amount;
        check minimum Balance ();
    }
}

public class Main {
    public static void main (String [] args) {
        Scanner sc = new Scanner (System.in);
        SavingsAccount s = new SavingsAccount ("customer Name", "account number: 1", initial balance: 1000.0);
        CurrentAccount currentAccount = new CurrentAccount ("Customer Name: Anmol B", "account number: 2", initial Balance: 5000.0);
    }
}

```



double withdrawalAmount = sc.nextDouble();
 if (SelectedAccount instanceof saving Account) {
 ((saving Account) selected Account).withdraw (withdrawal Amount);
 } else if (SelectedAccount instanceof current Account) {
 ((current Account) selected Account).withdraw (withdrawal Amount);
 }
 break;
}
Case 3:
if (Selected Account instanceof saving Account) {
 ((saving Account) selected Account).computeInterest();
} else {
 System.out.println("Interest computation is only for saving account");
}
break;
}
Case 4:
System.out.println("Account name: " + selected Account. accountName);
System.out.println("Account number: " + selected Account. account number);
System.out.println("Type of account: " + selected Account. accountType);
Selected Account. displayBalance();
break;
}
Case 5:
System.out.println("Exiting program....");
sc.close();
}



Code :

```
import java.util.Scanner;

class Account {
    String customerName;
    int accountNumber;
    String accountType;
    double balance;

    Account(String customerName, int accountNumber, String accountType,
    double initialBalance) {
        this.customerName = customerName;
        this.accountNumber = accountNumber;
        this.accountType = accountType;
        this.balance = initialBalance;
    }

    void deposit(double amount) {
        balance += amount;
        System.out.println("Deposit successful. New balance: $" + balance);
    }

    void displayBalance() {
        System.out.println("Current balance: $" + balance);
    }
}

class SavingsAccount extends Account {
    private static final double INTEREST_RATE = 0.04;

    SavingsAccount(String customerName, int accountNumber, double
    initialBalance) {
        super(customerName, accountNumber, "Savings", initialBalance);
    }

    void computeInterest() {
        double interest = balance * INTEREST_RATE;
        balance += interest;
        System.out.println("Interest added. New balance: $" + balance);
    }

    void withdraw(double amount) {
        if (amount > balance) {
```

```

        System.out.println("Insufficient funds.");
    } else {
        balance -= amount;
        System.out.println("Withdrawal successful. New balance: $" + balance);
    }
}

class CurrentAccount extends Account {
    private static final double MIN_BALANCE = 500.00;
    private static final double SERVICE_CHARGE = 25.00;

    CurrentAccount(String customerName, int accountNumber, double
initialBalance) {
        super(customerName, accountNumber, "Current", initialBalance);
    }

    void checkMinimumBalance() {
        if (balance < MIN_BALANCE) {
            balance -= SERVICE_CHARGE;
            System.out.println("Below minimum balance. Service charge of $" +
SERVICE_CHARGE + " applied. New balance: $" + balance);
        } else {
            System.out.println("Balance is above the minimum required.");
        }
    }

    void withdraw(double amount) {
        if (amount > balance) {
            System.out.println("Insufficient funds.");
        } else {
            balance -= amount;
            checkMinimumBalance();
        }
    }
}

public class Main {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        SavingsAccount savingsAccount = new SavingsAccount("Ar", 1,
1000.00);
    }
}

```

```

        CurrentAccount currentAccount = new CurrentAccount("Ayush", 2,
800.00);

        while (true) {
            System.out.println("\n---MENU---");
            System.out.println("1. Deposit");
            System.out.println("2. Withdraw");
            System.out.println("3. Compute interest for SavingsAccount");
            System.out.println("4. Display account details");
            System.out.println("5. Exit");
            System.out.print("Enter your choice: ");
            int choice = sc.nextInt();

            Account selectedAccount = null;
            System.out.print("Enter the type of account (saving/current): ");
            String accountType = sc.next();

            if (accountType.equalsIgnoreCase("saving")) {
                selectedAccount = savingsAccount;
            } else if (accountType.equalsIgnoreCase("current")) {
                selectedAccount = currentAccount;
            } else {
                System.out.println("Invalid account type. Try again.");
                continue;
            }

            switch (choice) {
                case 1:
                    System.out.print("Enter the deposit amount: ");
                    double depositAmount = sc.nextDouble();
                    selectedAccount.deposit(depositAmount);
                    break;
                case 2:
                    System.out.print("Enter the withdrawal amount: ");
                    double withdrawalAmount = sc.nextDouble();
                    if (selectedAccount instanceof SavingsAccount) {
                        ((SavingsAccount)
selectedAccount).withdraw(withdrawalAmount);
                    } else if (selectedAccount instanceof CurrentAccount) {
                        ((CurrentAccount)
selectedAccount).withdraw(withdrawalAmount);
                    }
                    break;
                case 3:
            }
        }
    }
}

```

```
if (selectedAccount instanceof SavingsAccount) {  
    ((SavingsAccount) selectedAccount).computeInterest();  
} else {  
    System.out.println("Interest computation is only for savings  
accounts.");  
  
}  
break;  
case 4:  
    System.out.println("Account name: " +  
selectedAccount.customerName);  
    System.out.println("Account number: " +  
selectedAccount.accountNumber);  
    System.out.println("Type of account: " +  
selectedAccount.accountType);  
    selectedAccount.displayBalance();  
    break;  
case 5:  
    System.out.println("Exiting program...");  
    sc.close();  
    return;  
default:  
    System.out.println("Invalid choice. Try again.");  
  
}  
}  
}  
}
```

Output:

---MENU---

1. Deposit
 2. Withdraw
 3. Compute interest for SavingsAccount
 4. Display account details
 5. Exit

Enter your choice: 1

Enter the type of account (saving/current): saving

Enter the deposit amount: 500

Deposit successful. New balance: \$1500.00

MENU

1. Deposit

- 2. Withdraw
- 3. Compute interest for SavingsAccount
- 4. Display account details
- 5. Exit

Enter your choice: 2

Enter the type of account (saving/current): current

Enter the withdrawal amount: 400

Withdrawal successful. New balance: \$400.0

Below minimum balance. Service charge of \$25.0 applied. New balance: \$375.0

---MENU---

- 1. Deposit
- 2. Withdraw
- 3. Compute interest for SavingsAccount
- 4. Display account details
- 5. Exit

Enter your choice: 3

Enter the type of account (saving/current): saving

Interest added. New balance: \$1560.0

---MENU---

- 1. Deposit
- 2. Withdraw
- 3. Compute interest for SavingsAccount
- 4. Display account details
- 5. Exit

Enter your choice: 4

Enter the type of account (saving/current): current

Account name: Ayush

Account number: 2

Type of account: Current

Current balance: \$375.0

---MENU---

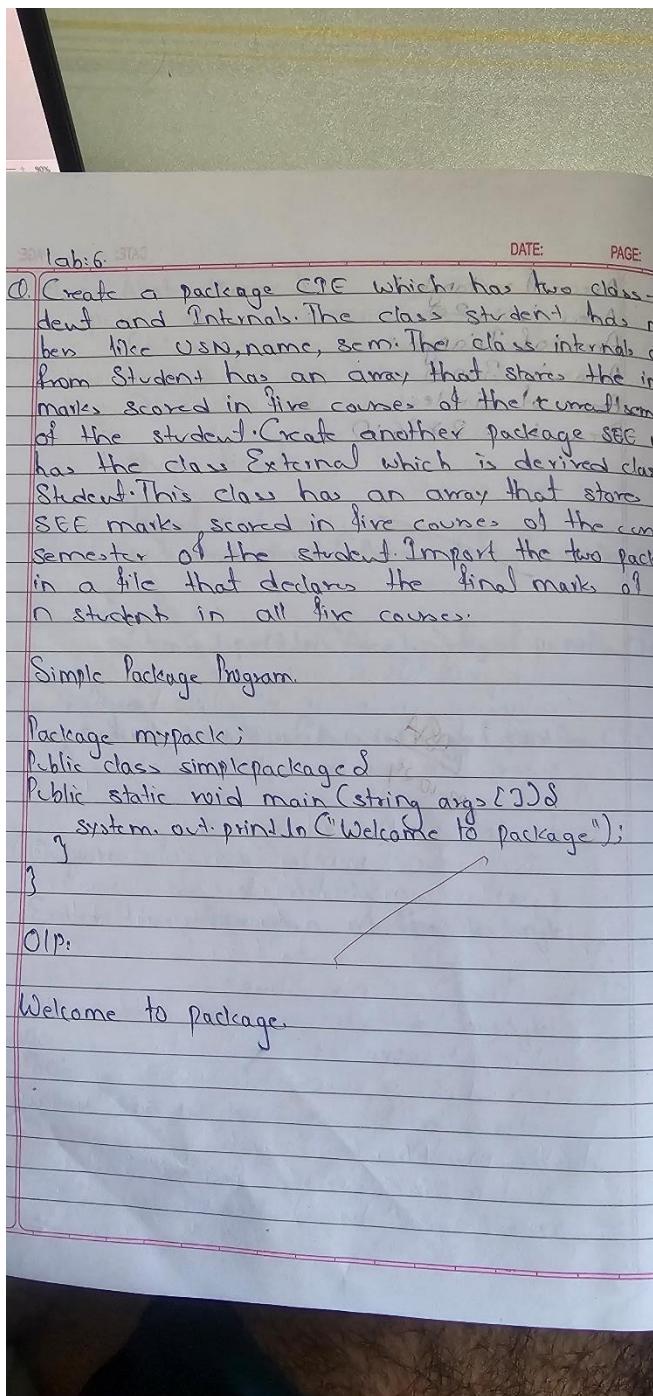
- 1. Deposit
- 2. Withdraw
- 3. Compute interest for SavingsAccount
- 4. Display account details
- 5. Exit

Enter your choice: 5

Exiting

Program No: 6
Implement Packages

Algorithm :



DATE: PAGE

```

import java.util.Scanner;
public class Main
{
    public static void main (String [] args)
    {
        Scanner scanner = new Scanner (System.in);
        System.out.print ("Enter the number of stds:");
        int n = scanner.nextInt ();
        scanner.nextLine ();
        for (int i = 0; i < n; i++)
        {
            System.out.println ("Enter detail for student" + i + ":");
            System.out.print ("Enter USN:");
            String usn = scanner.nextLine ();
            System.out.print ("Enter name:");
            String name = scanner.nextLine ();
            System.out.print ("Enter semest:");
            int sem = scanner.nextInt ();
            int [] internalMarks = new int [5];
            System.out.print ("Enter internal Marks for 5 courses:");
            for (int j = 0; j < 5; j++)
            {
                System.out.print ("course" + (j + 1) + ":");
                internalMarks [j] = scanner.nextInt ();
            }
            int [] externalMarks = new int [5];
            System.out.print ("Enter external marks for 5 courses:");
            for (int j = 0; j < 5; j++)
            {
                System.out.print ("course" + (j + 1) + ":");
                externalMarks [j] = scanner.nextInt ();
            }
        }
    }
}

```

```

Scanner.nextLine();
Internal internalStudent=new internal (usn, name,
internal Marks);
External externalStudent=new external (usn, name,
external Marks);
internal student.printInternalMarks();
external student.printExternalMarks();
printFinalMarks(internal student, external student);
Scanner.close();
public static void printFinalMarks(Internal internal, External external) {
int [] internalMarks = internal.internalMarks;
int [] externalMarks = external.externalMarks;
int [] totalMarks = internalMarks + externalMarks;
System.out.println("Final Marks for" + internal +
" (" + internal.getUsn() + "):");
for (int i = 0; i < internalMarks.length; i++) {
totalMarks[i] = internalMarks[i] + externalMarks[i];
System.out.println("course" + (i + 1) + ":" + totalMarks[i]);
}
}
package SEE;
import java.util.Scanner;
public class Internal extends Student {
public int [] internalMarks;
public internal (String usn, String name, int sem, int internalMarks) {
super (usn, name, sem);
this.internalMarks = internalMarks;
}
public void internalMarks() {
System.out.println("Internal Marks for" + name +
" " + usn + ":" );
for (int i = 0; i < internalMarks.length; i++) {
System.out.println("course" + (i + 1) + ":" + internalMarks[i]);
}
}
}
package SEE;
import java.util.Scanner;
public class External extends Student {
public int [] externalMarks;
public external (String usn, String name, int sem, int externalMarks) {
super (usn, name, sem);
this.externalMarks = externalMarks;
}
public void externalMarks() {
System.out.println("External Marks for" + name +
" " + usn + ":" );
for (int i = 0; i < externalMarks.length; i++) {
System.out.println("course" + (i + 1) + ":" + externalMarks[i]);
}
}
}

```

3048 7740 DATE

```
Public class student {  
    protected string USN;  
    protected string name;  
    protected string sem;  
    public Student(string usn, string name,  
    this.usn = usn,  
    this.name = name);  
    this.sem = sem;  
    public string getusn() {  
        return usn;  
    }  
    public string getname() {  
        return name;  
    }  
    public int getsem() {  
        return sem;  
    }  
}
```

OIP:

Enter no. of student: 1

Enter detail for student: 1

Enter USN: 1R2C

Enter name: ABCD

Enter sem: 3

Enter internal Marks for 5 subjects:

Sub 1: 80

Sub 2: 60

Sub 3: 90

Sub 4: 85

Sub 5: 70

Enter sec marks for 5 sub:

Sub 1: 70

Sub 2: 60

Sub 3: 75

Sub 4: 80

Sub 5: 90

Code:

```
package SEE;  
import CIE.Student;  
  
public class External extends Student {  
    public int[] externalMarks;
```

```

public External(String usn, String name, int sem, int[] externalMarks) {
    super(usn, name, sem);
    this.externalMarks = externalMarks;
}

public void printExternalMarks() {
    System.out.println("External Marks for " + name + " (" + usn + "):");
    for (int i = 0; i < externalMarks.length; i++) {
        System.out.println("Course " + (i + 1) + ": " + externalMarks[i]);
    }
}

package CIE;

public class Internals extends Student {
    public int[] internalMarks;

    public Internals(String usn, String name, int sem, int[] internalMarks) {
        super(usn, name, sem);
        this.internalMarks = internalMarks;
    }

    public void printInternalMarks() {
        System.out.println("Internal Marks for " + name + " (" + usn + "):");
        for (int i = 0; i < internalMarks.length; i++) {
            System.out.println("Course " + (i + 1) + ": " + internalMarks[i]);
        }
    }
}

import CIE.Internals;
import SEE.External;

import java.util.Scanner;

public class Main {
    public static void main(String[] args) {

```

```

Scanner scanner = new Scanner(System.in);

System.out.print("Enter number of students: ");
int n = scanner.nextInt();
scanner.nextLine();

for (int i = 0; i < n; i++) {

    System.out.println("\nEnter details for Student " + (i + 1) + ":");

    System.out.print("Enter USN: ");
    String usn = scanner.nextLine();

    System.out.print("Enter Name: ");
    String name = scanner.nextLine();

    System.out.print("Enter Semester: ");
    int sem = scanner.nextInt();

    int[] internalMarks = new int[5];
    System.out.println("Enter Internal Marks for 5 Courses:");
    for (int j = 0; j < 5; j++) {
        System.out.print("Course " + (j + 1) + ": ");
        internalMarks[j] = scanner.nextInt();
    }

    int[] externalMarks = new int[5];
    System.out.println("Enter External Marks for 5 Courses:");
    for (int j = 0; j < 5; j++) {
        System.out.print("Course " + (j + 1) + ": ");
        externalMarks[j] = scanner.nextInt();
    }
    scanner.nextLine();

    Internals internalStudent = new Internals(usn, name, sem, internalMarks);
    External externalStudent = new External(usn, name, sem, externalMarks);

    internalStudent.printInternalMarks();
    externalStudent.printExternalMarks();
}

```

```

        printFinalMarks(internalStudent, externalStudent);
    }

    scanner.close();
}

public static void printFinalMarks(Internals internal, External external) {
    int[] internalMarks = internal.internalMarks;
    int[] externalMarks = external.externalMarks;
    int totalMarks;

    System.out.println("Final Marks for " + internal.getName() + " (" + internal.getUsn() + ")");
    for (int i = 0; i < internalMarks.length; i++) {
        totalMarks = internalMarks[i] + externalMarks[i];
        System.out.println("Course " + (i + 1) + ": " + totalMarks);
    }
}

package CIE;

public class Student {
    protected String usn;
    protected String name;
    protected int sem;

    public Student(String usn, String name, int sem) {
        this.usn = usn;
        this.name = name;
        this.sem = sem;
    }

    // Getter methods
    public String getUsn() {
        return usn;
    }

    public String getName() {
        return name;
    }
}

```

```
public int getSem() {  
    return sem;  
}  
}
```

Output :

```
Enter the number of students: 2  
  
Enter details for student 1:  
USN: 18M23CS056  
Name: Ayaan Shrestha  
Semester: 3  
Enter internal marks for 5 courses:  
18 20 17 19 16  
Enter SEE marks for 5 courses:  
60 58 62 55 59  
  
Enter details for student 2:  
USN: 18M23CS058  
Name: Ayush Ranjan  
Semester: 3  
Enter internal marks for 5 courses:  
15 18 20 17 19  
Enter SEE marks for 5 courses:  
63 60 58 62 64
```

Final Marks of Students:

USN: IBM23CS056

Name: Ayaan Shrestha

Semester: 3

Internal Marks:

Course 1: 18

Course 2: 20

Course 3: 17

Course 4: 19

Course 5: 16

SEE Marks:

Course 1: 60

Course 2: 58

Course 3: 62

Course 4: 55

Course 5: 59

Final Marks:

Course 1: 78

Course 2: 78

Course 3: 79

Course 4: 74

Course 5: 75

USN: IBM23CS058

Name: Ayush Ranjan

Semester: 3

Internal Marks:

Course 1: 15

Course 2: 18

Course 3: 20

Course 4: 17

Course 5: 19

SEE Marks:

Course 1: 63

Course 2: 60

Course 3: 58

Course 4: 62

Course 5: 64

SRR Marks:

Course 1: 63

Course 2: 68

Course 3: 58

Course 4: 62

Course 5: 64

Final Marks:

Course 1: 78

Course 2: 78

Course 3: 78

Course 4: 79

Course 5: 83

Program No : 7
Implement Exception Handling

Algorithm :

Week 7 Lab Program

Q. WAP to demonstrate handling of exception in inheritance of tree create a base class called father and a derived class called son which extend class father. In father class implement a constructor which takes the age and throw the exception wrong age when input age < son's class implement a constructor that takes both father and son's age and through an exception if son's age \geq father's age (use)

```
import java.util.Scanner;
class WrongAgeException extends Exception {
    public WrongAgeException(String message) {
        super(message);
    }
}
class sonAgeException extends Exception {
    public sonAgeException(String message) {
        super(message);
    }
}
class father {
    private int age;
    public father(int age) throws WrongAgeException {
        if (age < 0) {
            throw new WrongAgeException("Age cannot be negative");
        } else if (age > 120) {
            throw new WrongAgeException("Age is too high");
        }
        this.age = age;
    }
    public int getAge() {
        return age;
    }
}
```

```

if (age < 0) {
    throw new WrongAgeException("Wrong Age");
}
this.age = age;
if (age < 0) {
    throw new WrongAgeException("Wrong Age");
}
public int getAge() {
    return age;
}
class Son extends Father {
    private int sonAge;
    public Son(int fatherAge, int sonAge) {
        throw new WrongAgeException("Wrong Age");
    }
    super(Father);
    if (sonAge >= fatherAge) {
        throw new SonAgeException("Son age > Father");
    }
    this.sonAge = sonAge;
}
public int getSonAge() {
    return sonAge;
}
public class FatherSon {
    public static void main(String[] args) {
        while (true)
    }
}

```

DATE: _____

8 (or 9) : 11

Scanner sc = new Scanner (System.in);
 System.out.println ("Father age");
 int f.age = sc.nextInt ();
 System.out.println ("Son age");
 int s.age = sc.nextInt ();

8 (or 10) : 11

try
 {
 Son s = new Son (f.age, s.age);
 System.out.println ("Accepted successfully");
} catch (WrongAgeException e) {
 System.out.println (e.getMessage());
}

8 (or 11) : 12

System.out.println ("(F/N) to enter");
 String s.input = sc.nextLine();
 if (input.equalsIgnoreCase ("N"))
 break;

8 (or 12) : 11

8 (or 13) : 11

8 (or 14) : 11

Code :

```
import java.util.Scanner;
class WrongAgeException extends Exception {
    public WrongAgeException(String message) {
        super(message);
    }
}

class SonAgeException extends Exception {
    public SonAgeException(String message) {
        super(message);
    }
}

class Father {
    private int age;
    public Father(int age) throws WrongAgeException {
        if (age < 0) {
            throw new WrongAgeException("Wrong age");
        }
        this.age = age;
    }
    public int getAge() {
        return age;
    }
}

class Son extends Father {
    private int sonAge;
    public Son(int fatherAge, int sonAge) throws WrongAgeException,
SonAgeException {
        super(fatherAge);
        if (sonAge >= fatherAge) {
            throw new SonAgeException("Son's age cannot be greater than or equal
to father's age");
        }
        this.sonAge = sonAge;
    }
    public int getSonAge() {
        return sonAge;
    }
}

public class FatherSon{
```

```

public static void main(String[] args) {
    while(true){
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter Father's Age: ");
        int fatherAge = sc.nextInt();
        System.out.print("Enter Son's Age: ");
        int sonAge = sc.nextInt();
        try {
            Son son = new Son(fatherAge, sonAge);
            System.out.println("Accepted Successfully");
        }
        catch (WrongAgeException e) {
            System.out.println(e.getMessage());
        }
        catch (SonAgeException e) {
            System.out.println(e.getMessage());
        }
        System.out.println("Would you like to re-enter details (Y/n)");
        String input = sc.next();
        if(input.equalsIgnoreCase("n")){
            break;
        }
    }
}
}

```

Output:

```

C:\Users\Admin\Desktop>javac FatherSon.java
C:\Users\Admin\Desktop>java FatherSon
Enter Father's Age: 51
Enter Son's Age: 19
Accepted Successfully
Would you like to re-enter details (Y/n)
Y
Enter Father's Age: 24
Enter Son's Age: 25
Son's age cannot be greater than or equal to father's age
Would you like to re-enter details (Y/n)
N

```

Program No : 8

Multithreading, Creating Threads in Java

Algorithm :

WAP that creates two threads, one thread displaying "BMS College of Engineering" once every ten seconds and another displaying "CSE" once every ten seconds.

```

class BMS extends Thread {
    public void run() {
        try {
            while (true) {
                System.out.println("BMS College of Engineering");
                Thread.sleep(10000);
            }
        } catch (InterruptedException e) {
        }
    }
}

class CSE extends Thread {
    public void run() {
        try {
            while (true) {
                System.out.println("CSE");
                Thread.sleep(2000);
            }
        } catch (InterruptedException e) {
        }
    }
}

public class Multithreading {
    public class void main (String [] args) {
        BMS bms = new BMS();
        CSE cse = new CSE();
        bms.start();
        cse.start();
    }
}

```

Output of the code is as follows:

Computer Science: 1st output

Computer Science: 2nd output

Computer Science: 3rd output

Computer Science: 4th output

Computer Science: 5th output

Information Science: 1st output

Information Science: 2nd output

Information Science: 3rd output

Information Science: 4th output

Information Science: 5th output

Information Science: 6th output

Information Science: 7th output

Information Science: 8th output

Information Science: 9th output

Information Science: 10th output

Code :

```
class BMS extends Thread {  
    public void run() {  
        try {  
            while (true) {
```

```

        System.out.println("BMS College of Engineering");
        Thread.sleep(10000); // Sleep for 10 seconds
    }
} catch (InterruptedException e) {}
}

class CSE extends Thread {
    public void run() {
        try {
            while (true) {
                System.out.println("CSE");
                Thread.sleep(2000); // Sleep for 2 seconds
            }
        } catch (InterruptedException e) {}
    }
}

public class Multithreading{
    public static void main(String[] args) {
        BMS bms = new BMS();
        CSE cse = new CSE();
        bms.start();
        cse.start();
    }
}

```

Output:

```

C:\Users\Admin\Desktop>java Multithreading
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
BMS College of Engineering
CSE
CSE
^C
C:\Users\Admin\Desktop>

```

Program No : 9

Interface to Perform Integer Division

Algorithm :

Lab Program 9

WAP that creates a user interface to perform division. The user enters two numbers in the fields, Num1, Num2. The division of Num1 and Num2 is displayed in the result field when the Divide button is clicked. If Num1 or Num2 were not an integer, the program would throw a NumberFormatException. If Num2 were zero, the program would throw an ArithmeticException. Display the exception message dialogue box.

```

import javax.swing.*;
import java.awt.*;
import java.awt.event.*;
class SwingDemo extends SwingDemo {
    SwingDemo() {
        JFrame jfrm = new JFrame("Divide App");
        jfrm.setSize(275, 180);
        jfrm.setLayout(new FlowLayout());
        jfrm.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        JLabel jlab = new JLabel("Enter divider");
        JTextField ajtf = new JTextField(8);
        JTextField bjtf = new JTextField(8);
        JButton button = new JButton("Calculate");
        JLabel err = new JLabel();
        JLabel alab = new JLabel();
        JLabel blab = new JLabel();
        JLabel anolab = new JLabel();
        jlab.setBounds(10, 10, 150, 30);
        ajtf.setBounds(10, 40, 150, 30);
        bjtf.setBounds(10, 70, 150, 30);
        button.setBounds(10, 100, 150, 30);
        err.setBounds(10, 130, 150, 30);
        alab.setBounds(10, 160, 150, 30);
        blab.setBounds(10, 190, 150, 30);
        anolab.setBounds(10, 220, 150, 30);
        jfrm.add(jlab);
        jfrm.add(ajtf);
        jfrm.add(bjtf);
        jfrm.add(button);
        jfrm.add(alab);
        jfrm.add(blab);
        jfrm.add(anolab);
    }
}

```

DATE: PAGE:

```

jfrm.add(err);
jfrm.add(alab);
jfrm.add(blab);
jfrm.add(anolab);
}
ActionListener I = new ActionListener() {
    public void actionPerformed(ActionEvent evt) {
        System.out.println("Action event from a text field");
    }
};
ajtf.addActionListener(I);
bjtf.addActionListener(I);
button.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent evt) {
        try {
            int a = Integer.parseInt(ajtf.getText());
            int b = Integer.parseInt(bjtf.getText());
            int ans = a/b;
            alab.setText("In A = " + a);
            blab.setText("In B = " + b);
            anolab.setText("In Ans = " + ans);
        } catch (NumberFormatException e) {
            alab.setText(" ");
            blab.setText(" ");
            anolab.setText(" ");
        }
    }
});

```


Code :

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;

class SwingDemo {
    SwingDemo() {
        JFrame jfrm = new JFrame("Divider App");
    }
}
```

```

jfrm.setSize(275, 150);
jfrm.setLayout(new FlowLayout());

jfrm.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

JLabel jlab = new JLabel("Enter the divisor and dividend:");

JTextField ajtf = new JTextField(8);
JTextField bjtf = new JTextField(8);

JButton button = new JButton("Calculate");

JLabel err = new JLabel();
JLabel alab = new JLabel();
JLabel blab = new JLabel();
JLabel anslab = new JLabel();

jfrm.add(err);
jfrm.add(jlab);
jfrm.add(ajtf);
jfrm.add(bjtf);
jfrm.add(button);
jfrm.add(alab);
jfrm.add(blab);
jfrm.add(anslab);

button.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent evt) {
        try {

            int a = Integer.parseInt(ajtf.getText());
            int b = Integer.parseInt(bjtf.getText());

            int ans = a / b;
            alab.setText("A = " + a);
            blab.setText("B = " + b);
            anslab.setText("Ans = " + ans);

        }
    }
});

```

```

        err.setText("");
    } catch (NumberFormatException e) {

        alab.setText("");
        blab.setText("");
        anslab.setText("");
        err.setText("Enter Only Integers!");
    } catch (ArithmaticException e) {

        alab.setText("");
        blab.setText("");
        anslab.setText("");
        err.setText("B should be NON-zero!");
    }
}
});

```

```

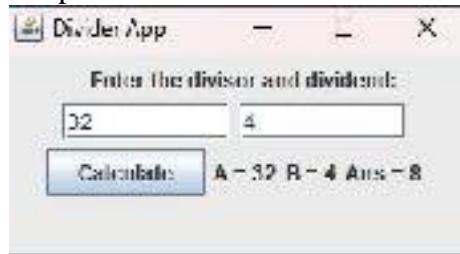
        jfrm.setVisible(true);
    }

public static void main(String args[]) {

    SwingUtilities.invokeLater(new Runnable() {
        public void run() {
            new SwingDemo();
        }
    });
}

```

Output :



Program 10.1

Implement Deadlock

Algorithm :

```
DATE: PAGE:  
→ Deadlock Program  
("L014" name) using lock interface  
Class A {  
    synchronized void foo(B b) {  
        String name = Thread.currentThread().getName();  
        Thread.currentThread().getName();  
        System.out.println(name + " entered A.foo");  
        try {  
            Thread.sleep(1000);  
        } catch (Exception e) {  
            System.out.println("A interrupted");  
        }  
        System.out.println("Inside A.foo");  
        b.last();  
    }  
    synchronized void last() {  
        System.out.println("Inside A.last");  
        try {  
            Thread.sleep(1000);  
        } catch (Exception e) {  
            System.out.println("A interrupted");  
        }  
        System.out.println("B interrupted");  
        b.foo();  
    }  
}  
Class B {  
    synchronized void bar(A a) {  
        String name = Thread.currentThread().getName();  
        Thread.currentThread().getName();  
        System.out.println(name + " entered B.bar");  
        try {  
            Thread.sleep(1000);  
        } catch (Exception e) {  
            System.out.println("B interrupted");  
        }  
        System.out.println("Inside B.bar");  
        a.last();  
    }  
}
```

DATE: PAGE:
 void last() {
 System.out.println("Inside A.last");
 }
 }
 class Deadlock implements Runnable {
 }
 A a = new A();
 B b = new B();
 Deadlock d = new Deadlock();
 Thread t1 = new Thread(d);
 Thread t2 = new Thread(this);
 t1.start();
 a.foo();
 System.out.println("Back in main thread");
 public void run() {
 b.bar();
 System.out.println("Back in other thread");
 }
 public static void main(String args[]) {
 new Deadlock();
 }
 O/P:
 MainThread entered A.foo()
 RacingThread entered B.bar()
 MainThread trying to call B.last()
 Inside A.last
 Back in main thread
 Racing Thread trying to call A.last()

DATE: 2020-01-01

Inside A. last
Back in other thread

O/P for Inter process communication.

Press control-c to stop

put = 0

Intimate consumer
producer waiting
got = 0

Intimate producer
put = 1

Intimate consumer
producer waiting

Consumed = 0
got = 1

Intimate producer
consumer = 1
put = 2

Intimate consumer
producer waiting
got = 2

DATE: 2020-01-01

Intimate producer
consumer = 2
put = ?

Intimate consumer
producer waiting
got = 3

Intimate producer
consumer = 3
put = 4

Intimate consumer
got = 4

Intimate producer
consumer = 4

Block

8 =

Code :

```
class A {  
    synchronized void foo(B b) {  
        String name = Thread.currentThread().getName();  
        System.out.println(name + " entered A.foo");  
  
        try {  
            Thread.sleep(1000);  
        } catch (Exception e) {
```

```

        System.out.println("A Interrupted");
    }

synchronized (b) {
    System.out.println(name + " trying to call B.last()");
    b.last();
}
}

void last() {
    System.out.println("Inside A.last");
}
}

class B {
    synchronized void bar(A a) {
        String name = Thread.currentThread().getName();
        System.out.println(name + " entered B.bar");

        try {
            Thread.sleep(1000);
        } catch (Exception e) {
            System.out.println("B Interrupted");
        }
    }

    synchronized (a) {
        System.out.println(name + " trying to call A.last()");
        a.last();
    }
}

void last() {
    System.out.println("Inside B.last");
}
}

class Deadlock implements Runnable {
    A a = new A();
    B b = new B();

    Deadlock() {
        Thread.currentThread().setName("MainThread");
    }
}

```

```

        Thread t = new Thread(this, "RacingThread");
        t.start();

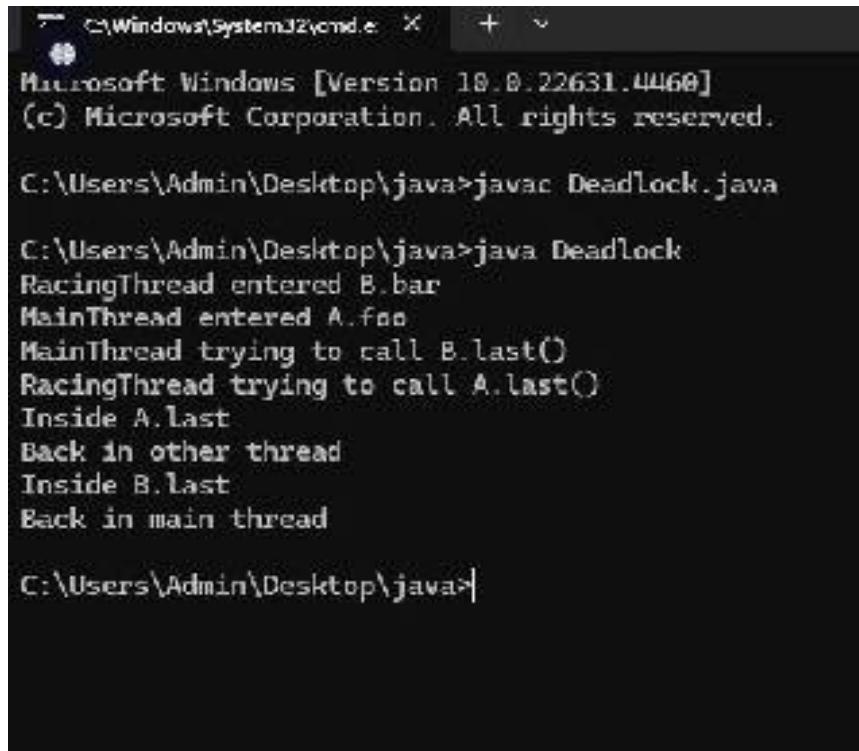
        a.foo(b);
        System.out.println("Back in main thread");
    }

    public void run() {
        b.bar(a);
        System.out.println("Back in other thread");
    }

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

```

Output:



```

C:\Windows\system32\cmd.exe  x  +  ~
Microsoft Windows [Version 10.0.22631.4460]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Admin\Desktop>java Deadlock.java

C:\Users\Admin\Desktop>java Deadlock
RacingThread entered B.bar
MainThread entered A.foo
MainThread trying to call B.last()
RacingThread trying to call A.last()
Inside A.last
Back in other thread
Inside B.last
Back in main thread

C:\Users\Admin\Desktop>

```

Program 10.2 :
Implement Inter-process Communication

Algorithm :

LAP program: 10
 ① Demonstrate Inter process Communication and deadlock

```

    DATE: PAGE:
    Class C {
        int n;
        boolean valueSet = false;
        synchronized int get() {
            while (!valueSet)
                try {
                    System.out.println("In Consumer Waiting");
                    wait();
                } catch (InterruptedException e) {
                    System.out.println("InterruptedException Caught");
                }
            System.out.println("Got: " + n);
            valueSet = true;
            System.out.println("In Intimate Producer");
            notify();
            return n;
        }
        synchronized void put(int n) {
            while (valueSet)
                try {
                    System.out.println("In Producer Waiting");
                    wait();
                } catch (InterruptedException e) {
                    System.out.println("InterruptedException Caught");
                }
            this.n = n;
            valueSet = true;
            System.out.println("Put: " + n);
            System.out.println("In Intimate Consumer");
        }
    }

```

```

    notify();
    } // end of run()
}

class producer implements Runnable {
    Queue<String> q;
    Producer<Queue<String> > producer;
    Consumer<Queue<String> > consumer;
    this.q = q;
    new Thread(this, "producer").start();
}

public void run() {
    int i = 0;
    while (i < 15) {
        q.put(i++);
    }
}

class consumer implements Runnable {
    Queue<String> q;
    Consumer<Queue<String> > consumer;
    this.q = q;
    new Thread(this, "consumer").start();
}

public void run() {
    int i = 0;
    while (i < 15) {
        int r = q.get();
        System.out.println("consumed: " + r);
        i += r;
    }
}

```

Class PCFixedS
 Public static void main (String args []) {
 Queue<String> q = new Queue<String>();
 new producer(q); // internal producer
 new consumer(q);
 System.out.println("Press Control-C to stop");
 }
}

class Producer {
 public void put (String s) {
 synchronized (this) {
 System.out.println("Producer: " + s);
 this.notify();
 }
 }
}

class Consumer {
 public void get (String s) {
 synchronized (this) {
 System.out.println("Consumer: " + s);
 this.notify();
 }
 }
}

Code :

```
class SharedResource {  
    private int data;  
    private boolean isDataAvailable = false;  
  
    public synchronized void produceData(int data) throws InterruptedException {  
        while (isDataAvailable) {  
            wait();  
        }  
        this.data = data;  
        System.out.println("Put: " + data);  
        isDataAvailable = true;  
        notify();  
    }  
  
    public synchronized void consumeData() throws InterruptedException {  
        while (!isDataAvailable) {  
            wait();  
        }  
        System.out.println("Get: " + data);  
        isDataAvailable = false;  
        notify();  
    }  
}  
  
public class IPCDemo {  
    public static void main(String[] args) {  
        SharedResource sharedResource = new SharedResource();  
  
        Thread producer = new Thread(() -> {  
            try {  
                for (int i = 0; i < 5; i++) {  
                    sharedResource.produceData(i);  
                    Thread.sleep(1000);  
                }  
            } catch (InterruptedException e) {  
                e.printStackTrace();  
            }  
        });  
    }  
}
```

```
Thread consumer = new Thread(() -> {
    try {
        for (int i = 0; i < 5; i++) {
            sharedResource.consumeData();
            Thread.sleep(1500);
        }
    } catch (InterruptedException e) {
        e.printStackTrace();
    }
});

producer.start();
consumer.start();
System.out.println("Anmol Bhattacharai 1BM23CS039");
}
}
```

OutPut:

```
CAWindows\System32\cmdc  X  |  ~

Producer waiting

Get: 11
Consumed: 11
Put: 12

Intimate Consumer

Producer waiting

Get: 12
Consumed: 12
Put: 13

Intimate Consumer

Producer waiting

Get: 13
Consumed: 13
Put: 14

Intimate Consumer

Get: 14
Consumed: 14

C:\Users\Admin\Desktop\java>
```