

K.R. Mangalam University

School of Engineering & Technology



Fundamentals Of Java Programming

Assignment 1

Banking Application for Accounts Management

Submitted by:

Name: Khushi

Roll No: 2401201102

Course: BCA (AI & DS)

Section: B

CODE:

```
1 import java.util.Scanner;
2
3 // Account Class Definition
4 class Account {
5     // fields
6     private final int accountNumber;
7     private String name, email, phone;
8     private double balance;
9
10    // constructor
11    Account(int accNo, String name, double balance, String email, String phone) {
12        this.accountNumber = accNo;
13        this.name = name;
14        this.balance = balance;
15        this.email = email;
16        this.phone = phone;
17    }
18
19    // method to get account number
20    public int getAccountNumber() {
21        return accountNumber;
22    }
23
24    // method to deposit money
25    void deposit(double amt) {
26        if (amt > 0) {
27            balance += amt;
28            System.out.println("Deposit Successful. Balance: " + balance);
29        } else {
30            System.out.println("Invalid deposit amount.");
31        }
32    }

```

```
33
34     // method to withdraw money
35     void withdraw(double amt) {
36         if (amt > 0 && amt <= balance) {
37             balance -= amt;
38             System.out.println("Withdrawal Successful. Balance: " + balance);
39         } else {
40             System.out.println("Invalid or insufficient balance.");
41         }
42     }
43
44     // method to display account details
45     void show() {
46         System.out.println("Account Number: " + accountNumber + " | Name: " + name +
47                             " | Balance: " + balance + " | Email: " + email +
48                             " | Phone: " + phone);
49     }
50
51     // method to update contact details
52     void update(String email, String phone) {
53         this.email = email;
54         this.phone = phone;
55         System.out.println("Contact updated!");
56     }
57 }
58
59 // Main Banking Application Class
60 public class BankingApp {
61 }
```

```
62 // Static members for the application state
63 static Scanner sc = new Scanner(System.in);
64 static Account[] accounts = new Account[100]; // Array to store up to 100 accounts
65 static int count = 0; // Counter for the number of accounts created
66
67 // Method to find an account by account number
68 static Account find(int accNo) {
69     for (int i = 0; i < count; i++) {
70         if (accounts[i].getAccountNumber() == accNo) {
71             return accounts[i];
72         }
73     }
74     return null; // Return null if not found
75 }
76
77 Run | Debug
78 public static void main(String[] args) {
79
80     while (true) {
81         // Display Menu (Customized for Khushi)
82         System.out.println("**** WELCOME TO KHUSHI'S BANKING APP ****");
83         System.out.println("1. Create Account");
84         System.out.println("2. Deposit");
85         System.out.println("3. Withdraw");
86         System.out.println("4. Show Account Details");
87         System.out.println("5. Update Contact Info");
88         System.out.println("6. Exit");
89         System.out.print("Enter your choice: ");
90
91         int choice = sc.nextInt();
92         sc.nextLine(); // consume newline
```

```
92
93     switch (choice) {
94         case 1 -> {
95             System.out.print(s: "Enter Name: ");
96             String name = sc.nextLine();
97             System.out.print(s: "Enter Initial Deposit: ");
98             double bal = sc.nextDouble();
99             sc.nextLine();
100            System.out.print(s: "Enter Email: ");
101            String email = sc.nextLine();
102            System.out.print(s: "Enter Phone: ");
103            String phone = sc.nextLine();
104
105            accounts[count] = new Account(1000 + count + 1, name, bal, email, phone);
106            System.out.println("Account Created: " + accounts[count].getAccountNumber());
107            count++;
108        }
109
110        case 2 -> {
111            System.out.print(s: "Acc No: ");
112            int accNo = sc.nextInt();
113            System.out.print(s: "Amount to deposit: ");
114            double amt = sc.nextDouble();
115
116            Account a = find(accNo);
117            if (a != null) {
118                a.deposit(amt);
119            } else {
120                System.out.println(x: "Not found.");
```

Activate Window
Go to Settings to act

```
121 }  
122 }  
123  
124 case 3 -> {  
125     System.out.print(s: "Acc No: ");  
126     int accNo = sc.nextInt();  
127     System.out.print(s: "Amount to withdraw: ");  
128     double amt = sc.nextDouble();  
129  
130     Account a = find(accNo);  
131     if (a != null) {  
132         a.withdraw(amt);  
133     } else {  
134         System.out.println(x: "Not found.");  
135     }  
136 }  
137  
138 case 4 -> {  
139     System.out.print(s: "Acc No: ");  
140     int accNo = sc.nextInt();  
141     Account a = find(accNo);  
142     if (a != null) {  
143         a.show();  
144     } else {  
145         System.out.println(x: "Not found.");  
146     }  
147 }  
148  
149 case 5 -> {  
150     System.out.print(s: "Acc No: ");  
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 }  
780 }  
781 }  
782 }  
783 }  
784 }  
785 }  
786 }  
787 }  
788 }  
789 }  
790 }  
791 }  
792 }  
793 }  
794 }  
795 }  
796 }  
797 }  
798 }  
799 }  
800 }  
801 }  
802 }  
803 }  
804 }  
805 }  
806 }  
807 }  
808 }  
809 }  
810 }  
811 }  
812 }  
813 }  
814 }  
815 }  
816 }  
817 }  
818 }  
819 }  
820 }  
821 }  
822 }  
823 }  
824 }  
825 }  
826 }  
827 }  
828 }  
829 }  
830 }  
831 }  
832 }  
833 }  
834 }  
835 }  
836 }  
837 }  
838 }  
839 }  
840 }  
841 }  
842 }  
843 }  
844 }  
845 }  
846 }  
847 }  
848 }  
849 }  
850 }  
851 }  
852 }  
853 }  
854 }  
855 }  
856 }  
857 }  
858 }  
859 }  
860 }  
861 }  
862 }  
863 }  
864 }  
865 }  
866 }  
867 }  
868 }  
869 }  
870 }  
871 }  
872 }  
873 }  
874 }  
875 }  
876 }  
877 }  
878 }  
879 }  
880 }  
881 }  
882 }  
883 }  
884 }  
885 }  
886 }  
887 }  
888 }  
889 }  
890 }  
891 }  
892 }  
893 }  
894 }  
895 }  
896 }  
897 }  
898 }  
899 }  
900 }  
901 }  
902 }  
903 }  
904 }  
905 }  
906 }  
907 }  
908 }  
909 }  
910 }  
911 }  
912 }  
913 }  
914 }  
915 }  
916 }  
917 }  
918 }  
919 }  
920 }  
921 }  
922 }  
923 }  
924 }  
925 }  
926 }  
927 }  
928 }  
929 }  
930 }  
931 }  
932 }  
933 }  
934 }  
935 }  
936 }  
937 }  
938 }  
939 }  
940 }  
941 }  
942 }  
943 }  
944 }  
945 }  
946 }  
947 }  
948 }  
949 }  
950 }  
951 }  
952 }  
953 }  
954 }  
955 }  
956 }  
957 }  
958 }  
959 }  
960 }  
961 }  
962 }  
963 }  
964 }  
965 }  
966 }  
967 }  
968 }  
969 }  
970 }  
971 }  
972 }  
973 }  
974 }  
975 }  
976 }  
977 }  
978 }  
979 }  
980 }  
981 }  
982 }  
983 }  
984 }  
985 }  
986 }  
987 }  
988 }  
989 }  
990 }  
991 }  
992 }  
993 }  
994 }  
995 }  
996 }  
997 }  
998 }  
999 }  
1000 }
```

```
151     int accNo = sc.nextInt();
152     sc.nextLine();
153     System.out.print(s: "New Email: ");
154     String email = sc.nextLine();
155     System.out.print(s: "New Phone: ");
156     String phone = sc.nextLine();
157
158     Account a = find(accNo);
159     if (a != null) {
160         a.update(email, phone);
161     } else {
162         System.out.println(x: "Not found.");
163     }
164 }
165
166 case 6 -> {
167     System.out.println(x: "Exiting. Thank you!");
168     return;
169 }
170
171 default -> System.out.println(x: "Invalid choice!");
172 }
```

OUTPUT:

```
*** WELCOME TO KHUSHI'S BANKING APP ***
1. Create Account
2. Deposit
3. Withdraw
4. Show Account Details
5. Update Contact Info
6. Exit
Enter your choice: 1
Enter Name: Khushi
Enter Initial Deposit: 5000
Enter Email: khushi@gmail.com
Enter Phone: 9876543210
Account Created: 1001

*** WELCOME TO KHUSHI'S BANKING APP ***
1. Create Account
2. Deposit
3. Withdraw
4. Show Account Details
5. Update Contact Info
6. Exit
Enter your choice: 2
Acc No: 1001
Amount to deposit: 2000
Deposit Successful. Balance: 7000.0
```

```
*** WELCOME TO KHUSHI'S BANKING APP ***
```

1. Create Account
2. Deposit
3. Withdraw
4. Show Account Details
5. Update Contact Info
6. Exit

```
Enter your choice: 3
```

```
Acc No: 1001
```

```
Amount to withdraw: 1500
```

```
Withdrawal Successful. Balance: 5500.0
```

```
*** WELCOME TO KHUSHI'S BANKING APP ***
```

1. Create Account
2. Deposit
3. Withdraw
4. Show Account Details
5. Update Contact Info
6. Exit

```
Enter your choice: 4
```

```
Acc No: 1001
```

```
Account Number: 1001 | Name: Khushi | Balance: 5500.0 | Email: khushi@gmail.com | Phone: 9876543210
```

```
*** WELCOME TO KHUSHI'S BANKING APP ***
```

1. Create Account
2. Deposit
3. Withdraw
4. Show Account Details
5. Update Contact Info
6. Exit

```
Enter your choice: 6
```

```
Exiting. Thank you!
```