

## Practice Programs

### Program:1

- Create a class Student with Name, Roll no and three subject marks as data member.
- Create a class result that inherit Student class. It has total and average as data member.
- Create a method calculate that will calculate the total marks of three subject with average of it.
- Use necessary method and constructor to initialize the data members.

```
import java.util.*;
class Student
{
    String name;
    int roll_no;
    double sub1_marks,sub2_marks,sub3_marks;
    Student()
    {
        Scanner sc=new Scanner(System.in);
        System.out.print("Enter Student Name : ");
        name=sc.nextLine();
        System.out.print("Enter Student Roll.no : ");
        roll_no=sc.nextInt();
        System.out.println();
        System.out.println("\n...Enter Marks Out Of 100....\n");
        System.out.println();
        System.out.print("Marks Of Subject-1 : ");
        sub1_marks=sc.nextDouble();
        System.out.print("Marks Of Subject-2 : ");
        sub2_marks=sc.nextDouble();
        System.out.print("Marks Of Subject-3 : ");
        sub3_marks=sc.nextDouble();
        System.out.println();
    }
}
class Result extends Student
{
    double total_marks;
    double avg_marks;
    void calculate(Student R)
    {
        total_marks=R.sub1_marks+R.sub2_marks+R.sub3_marks;
        avg_marks=total_marks/3;
    }
}
```

```

        }
    void display()
    {
        System.out.println("Total Marks = "+total_marks);
        System.out.println("Average Marks = "+avg_marks);
    }
}
class Main
{
    public static void main(String [] args)
    {
        Student s1=new Student();
        Result r1=new Result();
        r1.calculate(s1);
        r1.display();
    }
}

```

**Output:**

```

/*
E:\Ankur Patel\Sem-I\JAVA\T3 mcq>java Main
Enter Student Name : Hridank
Enter Student Roll.no : 28

```

"....Enter Marks Out Of 100...."

Marks Of Subject-1 : 98  
 Marks Of Subject-2 : 97  
 Marks Of Subject-3 : 99

Total Marks = 294.0  
 Average Marks = 98.0 \*/

---

**Programe:2**

- Declare a class called employee having employee\_id and employee\_name as members.
  - Extend class employee to have a subclass called salary having designation and monthly\_salary as members.
  - Define following: -
- 1) Required constructors - A method to find and display all details of employees drawing salary more than Rs. 20000/-.
  - 2) Method main for creating an array for storing these details given as an input and showing usage of above methods.(Use Scanner class).

```
import java.util.*;

class Employee
{
    Scanner sc=new Scanner(System.in);
    int employee_id;
    String employee_name;
}

class Salary extends Employee
{
    String designation;
    double monthly_salary;
    Salary()
    {
        System.out.println();
        System.out.print("employee_id = ");
        employee_id=sc.nextInt();
        sc.nextLine();
        System.out.print("employee_Name = ");
        employee_name=sc.nextLine();
        System.out.print("designation = ");
        designation=sc.nextLine();
        System.out.print("monthly_salary = ");
        monthly_salary=sc.nextDouble();
        System.out.println();
    }
    void display()
    {
        if(monthly_salary>20000)
        {
            System.out.println("employee_id = "+employee_id);
            System.out.println("employee_Name = "+employee_name);
            System.out.println("designation = "+designation);
            System.out.println("monthly_salary = "+monthly_salary);
            System.out.println();
        }
    }
}
class Main
{
    public static void main(String [] args)
    {
```

```
Scanner sc=new Scanner(System.in);
System.out.print("Enter no of Employees : ");
int n=sc.nextInt();
System.out.println();
Salary [] s=new Salary[n];
System.out.println("\n....Enter Following Details of Employees....\"");
System.out.println();
for(int i=0;i<n;i++)
{
    System.out.print("Enter Detail of Employee::["+(i+1)+"]::");
    System.out.println();
    s[i]=new Salary();
}
System.out.println("\n....Following Are the details of the Employees drawing
salary more than 20000....\"");
System.out.println();
for(int i=0;i<n;i++)
{
    s[i].display();
}
}
```

**Output:**

```
/*E:\Ankur Patel\Sem-I\JAVA\T3 mcq>java Main
Enter no of Employees : 4
```

"....Enter Following Details of Employees...."

**Enter Detail of Employee::[1]::**

```
employee_id = 11
employee_Name = Chagan
designation = Fitter
monthly_salary = 11000
```

**Enter Detail of Employee::[2]::**

```
employee_id = 12
employee_Name = Magan
designation = Junior Engineer
monthly_salary = 19500
```

Enter Detail of Employee::[3]::

employee\_id = 13  
employee\_Name = Bhuvan  
designation = Senior Engineer  
monthly\_salary = 31000

Enter Detail of Employee::[4]::

employee\_id = 14  
employee\_Name = Chaman  
designation = Maintanance Managar  
monthly\_salary = 49500

"....Following Are the details of the Employees drawing salary more than 20000..."

employee\_id = 13  
employee\_Name = Bhuvan  
designation = Senior Engineer  
monthly\_salary = 31000.0

employee\_id = 14  
employee\_Name = Chaman  
designation = Maintanance Managar  
monthly\_salary = 49500.0\*/

---

### Program:3

- Write a JAVA program to create a super class called figure that stores the dimensions of a two-dimensional object.
- It also defines a method called area () that computes the area of an object.
- The program derives two sub classes from figure.
- The first is rectangle and the second is Triangle.
- Each of these subclasses overrides area (),so that it returns the area of a rectangle and a triangle respectively

```
class Figure
{
    static double radius_of_circle;
    static double length_of_rectangle;
    static double width_of_rectangle;
    static double base_of_triangle;
    static double altitude_of_triangle;
```

```

void get(double R,double L,double W,double B,double A)
{
    radius_of_circle=R;
    length_of_rectangle=L;
    width_of_rectangle=W;
    base_of_triangle=B;
    altitude_of_triangle=A;
}

double area()
{
    double area_of_circle=3.14*radius_of_circle*radius_of_circle;
    return area_of_circle;
}
}

class Rectangle extends Figure
{

double area()
{
    double area_of_rectangle=length_of_rectangle*width_of_rectangle;
    return area_of_rectangle;
}
}

class Triangle extends Figure
{
double area()
{
    double area_of_triangle=0.5*base_of_triangle*altitude_of_triangle;
    return area_of_triangle;
}
}

class Main
{
    public static void main(String [] args)
    {
        Figure f1=new Figure();
        f1.get(10,10,20,20,10);
        System.out.println("area_of_circle="+f1.area());
        f1=new Rectangle();
        System.out.println("area_of_rectangle="+f1.area());
        f1=new Triangle();
        System.out.println("area_of_triangle="+f1.area());
    }
}

```

```
    }  
}
```

**Output:**

```
/*E:\Ankur Patel\Sem-I\JAVA\T3 mcq>java Main  
area_of_circle=314.0  
area_of_rectangle=200.0  
area_of_triangle=100.0*/
```

/\*Write a JAVA program as per given instruction.  
Create a class named "Student" in which you have to store data of students in  
instance variables named "Rollno" , "Name" , "Percentage".  
Create necessary methods to perform following operations in "Student" class.

1. "set" -- for assigning data of Students.
2. "get" (without return type & without argument) -- for displaying data of individual Students.
3. "get" (without return type & with argument) -- for displaying data of all Students.
4. "sort" -- for sorting data of Students based on their Rollno in ascending order without using inbuilt method.
5. "find" -- for searching data of Students based on their Rollno.  
(Take Rollno from user using scanner class and if Rollno exist then display details using get() method else print message "Record Not Found!!")
6. "update" -- for updating the name of a Student to Upper case based on their Rollno.  
(Take Rollno from user using scanner class and if Rollno exist then Update name to Upper case and then display details of all student using get() method else display old record)

\*\* Note: You have to create methods with the name same as given above and set category of "get" methods as per instruction,  
for remaining methods you can set as per your understanding.

-- Create another class named "RUN" which includes the main method. (use array of objects concept.)  
(take size of array through command line argument), call methods of "Student" class from this class.

```
/*
import java.util.*;
class Student
{
    Scanner sc = new Scanner (System.in);
    //instance Variable
    int Rollno;
    String Name;
    double Percentage;
    int i,j;
    // "set" -- for assigning data of Students.
    void set()
    {
        System.out.println("Enter roll no");
        Rollno = sc.nextInt();
        sc.nextLine();
        System.out.println("Enter Name");
        Name = sc.nextLine();
        System.out.println("Enter Percentage");
        Percentage = sc.nextDouble();
    }
    // "get" (without return type & with argument) -- for displaying data of
    all Students.
```

```

void get(Student arr[])
{
    for(i=0;i<arr.length;i++)
    {
        System.out.print("Roll no= "+arr[i].Rollno+" ");
        System.out.print("Name= "+arr[i].Name+" ");
        System.out.print("Percentage= "+arr[i].Percentage);
        System.out.println();
    }
}

// "get" (without return type & without argument) -- for displaying data
of individual Students.

void get()
{
    System.out.print("Roll no= "+Rollno+" ");
    System.out.print("Name= "+Name+" ");
    System.out.print("Percentage= "+Percentage);
    System.out.println();
}

// "sort" -- for sorting data of Students based on their Rollno in
ascending order without using inbuilt method.

void sort(Student arr[])
{
    Student temp = new Student();
    for(i=0;i<arr.length;i++)
    {
        for(j=i+1;j<arr.length;j++)
        {
            if(arr[i].Rollno>arr[j].Rollno)
            {
                temp = arr[i];
                arr[i]=arr[j];
                arr[j]=temp;
            }
        }
    }
}

// "find" -- for searching data of Students based on their Rollno.

int find()
{
    System.out.print("Enter roll no you want to find");
    int x = sc.nextInt();
    return x;
}

// "update" -- for updating the name of a Student to Upper case based on
their Rollno.

void update(Student arr[])
{
    System.out.print("Enter roll no for which you want to update
name");
    int dl = sc.nextInt();
    int count=0;
    for( i=0;i<arr.length;i++)
    {
}

```

```

        if(dl==arr[i].Rollno)
        {
            arr[i].Name=
                get(arr);
        }
        else
        {
            count++;
        }
    }
    if(count==arr.length)
    {
        get(arr);
    }
}
class RUN
{
    public static void main(String args[])
    {
        int Size = Integer.parseInt(args[0]); // Input of Size of
array through command line argument
        int count=0;
        int i;
        Student arr[] = new Student[Size];           // Array of
Object is created

        for(i=0;i<Size;i++)
        {
            arr[i]=new Student();
            arr[i].set();                         // Calling set method
for value assigning
        }

        Student ob = new Student();   // Object of class
"Student" is created

        ob.sort(arr);                      // sorting of array based
on Rollno

        ob.get(arr);                      // displaying data after
sorting

        int x = ob.find();                // call of "find" method
        for(i=0;i<Size;i++)
        {
            if(x==arr[i].Rollno)
            {
                arr[i].get();
            }
            else
            {

```

```
        count++;
    }
}
if(count==Size)
{
    System.out.println("Not found!!");
}
ob.update(arr); // call of update method
}
}
```

```

1  /*Create a mini-application in cmd for a banking system in Java and add some
2   basic functionalities of a bank account.
3
4   Initially, the program accepts the number of customers (n size for array of
5   customer objects) we need to add and adds
6   the customer and account details accordingly(use constructor). Further, it
7   displays the series of menus (Switch case) to operate over the accounts.
8
9   The series of menus displayed are as follows:
10
11  1.Display all account details
12  2.Search by account number
13  3.Deposit the amount
14  4.Withdraw the amount
15  5.Exit
16
17 Create required methods to fullfill above selections.
18
19 Hint:
20 Make class BankDetails having String accno, String name, String acc_type, long
21 balance.
22 In class Create constructor to open new account to scan details from user.
23 Create method to display account details public void showAccount()
24 Create method to deposit money public void deposit()
25 Create method to withdraw money public void withdrawal()
26 Create method to search customer by an account number public boolean
27 search(String ac_no)
28 to display him/her.
29
30 Create Main class to run main method and provide selection(switch case).*/
31
32 import java.util.Scanner;
33 class BankDetails {
34     String accno;
35     String name;
36     String acc_type;
37     long balance;
38     Scanner sc = new Scanner(System.in);
39     //method to open new account
40     BankDetails() {
41         System.out.print("Enter Account No: ");
42         accno = sc.next();
43         System.out.print("Enter Account type: ");
44         acc_type = sc.next();
45         System.out.print("Enter Name: ");
46         name = sc.next();
47         System.out.print("Enter Balance: ");
48         balance = sc.nextLong();
49     }
50     //method to display account details
51     public void showAccount() {
52         System.out.println("Name of account holder: " + name);
53         System.out.println("Account no.: " + accno);
54         System.out.println("Account type: " + acc_type);
55         System.out.println("Balance: " + balance);
56     }
57     //method to deposit money
58     public void deposit() {
59         long amt;
60         System.out.println("Enter the amount you want to deposit: ");
61         amt = sc.nextLong();
62         balance = balance + amt;
63     }
64     //method to withdraw money
65     public void withdrawal() {
66         long amt;

```

```

67     System.out.println("Enter the amount you want to withdraw: ");
68     amt = sc.nextLong();
69     if (balance >= amt) {
70         balance = balance - amt;
71         System.out.println("Balance after withdrawal: " + balance);
72     } else {
73         System.out.println("Your balance is less than " + amt +
74             "\tTransaction failed...!!!");
75     }
76 //method to search an account number
77 public boolean search(String ac_no) {
78     if (accno.equals(ac_no)) {
79         showAccount();
80         return (true);
81     }
82     return (false);
83 }
84 }
85 class BankingApp {
86     public static void main(String arg[]) {
87         Scanner sc = new Scanner(System.in);
88         //create initial accounts
89         System.out.print("How many number of customers do you want to input? ");
90         int n = sc.nextInt();
91         BankDetails C[] = new BankDetails[n];
92         for (int i = 0; i < C.length; i++) {
93             C[i] = new BankDetails();
94         }
95         // add initial customers and go to menu
96         // loop runs until number 5 is not pressed to exit
97         int ch;
98         do {
99             System.out.println("\n ***Banking System Application***");
100            System.out.println("1. Display all account details \n 2. Search by
101            Account number\n 3. Deposit the amount \n 4. Withdraw the amount \n
102            5.Exit ");
103            System.out.println("Enter your choice: ");
104            ch = sc.nextInt();
105            switch (ch) {
106                case 1:
107                    for (int i = 0; i < C.length; i++) {
108                        C[i].showAccount();
109                    }
110                    break;
111                case 2:
112                    System.out.print("Enter account no. you want to search: ");
113                    String ac_no = sc.next();
114                    boolean found = false;
115                    for (int i = 0; i < C.length; i++) {
116                        found = C[i].search(ac_no);
117                        if (found) {
118                            break;
119                        }
120                    }
121                    if (!found) {
122                        System.out.println("Search failed! Account doesn't
123                        exist..!!!");
124                    }
125                    break;
126                case 3:
127                    System.out.print("Enter Account no. : ");
128                    ac_no = sc.next();
129                    found = false;
130                    for (int i = 0; i < C.length; i++) {
131                        found = C[i].search(ac_no);
132                        if (found) {
133                            C[i].deposit();
134                            break;
135                        }
136                    }
137                case 4:
138                    System.out.print("Enter Account no. : ");
139                    ac_no = sc.next();
140                    found = false;
141                    for (int i = 0; i < C.length; i++) {
142                        found = C[i].search(ac_no);
143                        if (found) {
144                            C[i].withdraw();
145                            break;
146                        }
147                    }
148                case 5:
149                    System.out.println("Thank you for using our services");
150                    System.exit(0);
151            }
152        }
153    }

```

```
        }
    }
    if (!found) {
        System.out.println("Search failed! Account doesn't
exist..!!");
    }
    break;
case 4:
    System.out.print("Enter Account No : ");
    ac_no = sc.next();
    found = false;
    for (int i = 0; i < C.length; i++) {
        found = C[i].search(ac_no);
        if (found) {
            C[i].withdrawal();
            break;
        }
    }
    if (!found) {
        System.out.println("Search failed! Account doesn't
exist..!!");
    }
    break;
case 5:
    System.out.println("See you soon...");
```

break;

```
    }
}
while (ch != 5);
```

```
}
```

You are tasked to create a program that simulates computer parts using Java. The program should have the following classes:

1. ComputerPart (class) - This class should have the following properties:

- o partName (String) - the name of the computer part
- o partNumber (int) - the part number of the computer part
- o manufacturer (String) - the manufacturer of the computer part
- o price (double) - the price of the computer part

This class should also have the following methods:

- o ComputerPart(String partName, int partNumber, String manufacturer, double price) - a constructor that initializes the properties of the class
- o public void displayPartInfo() – a public method that displays the information of the computer part

2. CPU (subclass of ComputerPart) - This class should have the following properties:

- o coreCount (int) - the number of cores of the CPU
- o clockSpeed (double) - the clock speed of the CPU

This class should also have the following methods:

- o CPU(String partName, int partNumber, String manufacturer, double price, int coreCount, double clockSpeed) - a constructor that initializes the properties of the class using the constructor of the superclass (ComputerPart)
- o public void displayPartInfo() - an implementation of the abstract method of the superclass that displays the information of the CPU

3. GPU (subclass of ComputerPart) - This class should have the following properties:

- o memorySize (int) - the memory size of the GPU

This class should also have the following methods:

- o GPU(String partName, int partNumber, String manufacturer, double price, int memorySize) - a constructor that initializes the properties of the class using the constructor of the superclass (ComputerPart)
- o public void displayPartInfo() - an implementation of the method of the superclass that displays the information of the GPU

4. Computer (class) - This class should have the following properties:

- o cpu (CPU) - an instance of the CPU class
- o gpu (GPU) - an instance of the GPU class

This class should also have the following methods:

- o Computer(CPU cpu, GPU gpu) - a constructor that initializes the properties of the class
- o public void displayComputerInfo() - a method that displays the information of the computer, including the information of the CPU and the GPU

This class should also have the following inner class:

- o Keyboard - This class should have the following properties:
  - o language (String) - the language of the keyboard

This class should also have the following methods:

- o Keyboard(String language) - a constructor that initializes the properties of the class
- o public void displayKeyboardInfo() - a method that displays the information of the keyboard

The main method of the program should create an instance of the Computer class, and use the displayComputerInfo() method to display the information of the computer. It should also create an instance of the Keyboard class (inside the Computer class) and use the displayKeyboardInfo() method to display the information of the keyboard.

Design a class hierarchy for a library management system:

- Base class: Book
  - o Properties: ISBN, title, author, publicationDate, price
  - o Method: displayBookDetails()
  - o Constructor: Book(String ISBN, String title, String author, String publicationDate, double price)
  - o Use the "this" keyword to refer to instance variables in the class.
- Subclass: Novel extends Book
  - o Properties: genre
  - o Method: displayNovelDetails()
  - o Constructor: Novel(String ISBN, String title, String author, String publicationDate, double price, String genre)
  - o Use the "super" keyword to call the parent class constructor and access parent class properties.
- Subclass: Textbook extends Book
  - o Properties: subject, classNumber
  - o Method: displayTextbookDetails()

- Constructor: Textbook(String ISBN, String title, String author, String publicationDate, double price, String subject, int classNumber)
- Use the "super" keyword to call the parent class constructor and access parent class properties.
- Nested class: Borrower
  - Properties: borrowerID, borrowerName, address, phoneNumber
  - Method: displayBorrowerDetails()
  - Use the nested class to manage the borrower information for each book.
- Use the "final" keyword to prevent modification of the price in the Book class.
- Use constructor overloading to provide multiple ways to create a book object.

Use the "static" keyword to define a static variable to keep track of the total number of books in the library.

```
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
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
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
709
710
711
712
713
714
715
716
717
718
719
719
720
721
722
723
724
725
726
727
728
729
729
730
731
732
733
734
735
736
737
738
739
739
740
741
742
743
744
745
746
747
748
749
749
750
751
752
753
754
755
756
757
758
759
759
760
761
762
763
764
765
766
767
768
769
769
770
771
772
773
774
775
776
777
778
779
779
780
781
782
783
784
785
786
787
787
788
789
789
790
791
792
793
794
795
796
797
797
798
799
799
800
801
802
803
804
805
806
807
808
809
809
810
811
812
813
814
815
816
817
817
818
819
819
820
821
822
823
824
825
826
827
827
828
829
829
830
831
832
833
834
835
836
837
837
838
839
839
840
841
842
843
844
845
846
846
847
848
848
849
849
850
851
852
853
854
855
856
856
857
858
858
859
859
860
861
862
863
864
865
865
866
867
867
868
868
869
869
870
871
872
873
874
875
875
876
876
877
877
878
878
879
879
880
881
882
883
884
885
885
886
886
887
887
888
888
889
889
890
891
892
893
893
894
894
895
895
896
896
897
897
898
898
899
899
900
900
901
901
902
902
903
903
904
904
905
905
906
906
907
907
908
908
909
909
910
910
911
911
912
912
913
913
914
914
915
915
916
916
917
917
918
918
919
919
920
920
921
921
922
922
923
923
924
924
925
925
926
926
927
927
928
928
929
929
930
930
931
931
932
932
933
933
934
934
935
935
936
936
937
937
938
938
939
939
940
940
941
941
942
942
943
943
944
944
945
945
946
946
947
947
948
948
949
949
950
950
951
951
952
952
953
953
954
954
955
955
956
956
957
957
958
958
959
959
960
960
961
961
962
962
963
963
964
964
965
965
966
966
967
967
968
968
969
969
970
970
971
971
972
972
973
973
974
974
975
975
976
976
977
977
978
978
979
979
980
980
981
981
982
982
983
983
984
984
985
985
986
986
987
987
988
988
989
989
990
990
991
991
992
992
993
993
994
994
995
995
996
996
997
997
998
998
999
999
1000
1000
```

```
53 class Computer {
54     private CPU cpu;
55     private GPU gpu;
56
57     public Computer(CPU cpu, GPU gpu) {
58         this.cpu = cpu;
59         this.gpu = gpu;
60     }
61
62     public void displayComputerInfo() {
63         System.out.println("CPU Info:");
64         cpu.displayPartInfo();
65         System.out.println("GPU Info:");
66         gpu.displayPartInfo();
67     }
68
69     public class Keyboard {
70         private String language;
71
72         public Keyboard(String language) {
73             this.language = language;
74         }
75
76         public void displayKeyboardInfo() {
77             System.out.println("Keyboard Language: " + language);
78         }
79     }
80 }
81
82 public class Main {
83     public static void main(String[] args) {
84         CPU cpu = new CPU("Intel Core i7", 12345, "Intel", 80000,
85         );
86         GPU gpu = new GPU("Nvidia GeForce GTX 1080", 67890, "Nvidia",
87         110000, $);
88         Computer computer = new Computer(cpu, gpu);
89         computer.displayComputerInfo();
90
91         Computer.Keyboard keyboard = computer.new Keyboard("English");
92         keyboard.displayKeyboardInfo();
93     }
}
```

```
class Book {
    private final String ISBN;
    private final String title;
    private final String author;
    private final String publicationDate;
    private final double price;
    private static int totalBooks;

    public Book(String ISBN, String title, String author, String publicationDate, double price) {
        this.ISBN = ISBN;
        this.title = title;
        this.author = author;
        this.publicationDate = publicationDate;
        this.price = price;
        totalBooks++;
    }

    public static int getTotalBooks() {
        return totalBooks;
    }

    public void displayBookDetails() {
        System.out.println("ISBN: " + ISBN);
        System.out.println("Title: " + title);
        System.out.println("Author: " + author);
        System.out.println("Publication Date: " + publicationDate);
        System.out.println("Price: $" + price);
    }
}

class Novel extends Book {
    private final String genre;

    public Novel(String ISBN, String title, String author, String publicationDate, double price, String genre) {
        super(ISBN, title, author, publicationDate, price);
        this.genre = genre;
    }

    public void displayNovelDetails() {
        super.displayBookDetails();
        System.out.println("Genre: " + genre);
    }
}

class Textbook extends Book {
    private final String subject;
    private final int classNumber;

    public Textbook(String ISBN, String title, String author, String publicationDate, double price, String subject, int classNumber) {
        super(ISBN, title, author, publicationDate, price);
        this.subject = subject;
        this.classNumber = classNumber;
    }
}
```

```
53 }
54
55     public void displayTextbookDetails() {
56         super.displayBookDetails();
57         System.out.println("Subject: " + subject);
58         System.out.println("Class Number: " + classNumber);
59     }
60 }
61
62     class BookBorrower {
63         private final String borrowerID;
64         private final String borrowerName;
65         private final String address;
66         private final String phoneNumber;
67
68         public BookBorrower(String borrowerID, String borrowerName, String
address, String phoneNumber) {
69             this.borrowerID = borrowerID;
70             this.borrowerName = borrowerName;
71             this.address = address;
72             this.phoneNumber = phoneNumber;
73         }
74
75         public void displayBorrowerDetails() {
76             System.out.println("Borrower ID: " + borrowerID);
77             System.out.println("Borrower Name: " + borrowerName);
78             System.out.println("Address: " + address);
79             System.out.println("Phone Number: " + phoneNumber);
80     }
81 }
```