

Exercise 1:

```
7 *****
8 // Logger class
9 class Logger {
10     private static Logger instance;
11
12     private Logger() {
13         System.out.println("Logger initialized.");
14     }
15
16     public static Logger getInstance() {
17         if (instance == null) {
18             instance = new Logger();
19         }
20         return instance;
21     }
22
23     public void log(String message) {
24         System.out.println("Log: " + message);
25     }
26 }
27
28 // Main class
29 public class Main {
30     public static void main(String[] args) {
31         Logger logger1 = Logger.getInstance();
32         logger1.log("First log message");
33
34         Logger logger2 = Logger.getInstance();
35         logger2.log("Second log message");
36
37         if (logger1 == logger2) {
38             System.out.println("Only one Logger instance exists.");
39         } else {
40             System.out.println("Different Logger instances exist.");
41         }
42     }
43 }
44
```

Logger initialized.
Log: First log message
Log: Second log message
Only one Logger instance exists.
...Program finished with exit code 0
Press ENTER to exit console.

Exercise 2:

```
34 abstract class DocumentFactory {
35     public abstract Document createDocument();
36 }
37
38 // Concrete factories
39 class WordFactory extends DocumentFactory {
40     public Document createDocument() {
41         return new WordDocument();
42     }
43 }
44
45 class PdfFactory extends DocumentFactory {
46     public Document createDocument() {
47         return new PdfDocument();
48     }
49 }
50
51 class ExcelFactory extends DocumentFactory {
52     public Document createDocument() {
53         return new ExcelDocument();
54     }
55 }
56
57 // Main class to test Factory Pattern
58 public class Main {
59     public static void main(String[] args) {
60         DocumentFactory factory;
61
62         factory = new WordFactory();
63         Document doc1 = factory.createDocument();
64         doc1.open();
65
66         factory = new PdfFactory();
67         Document doc2 = factory.createDocument();
68         doc2.open();
69
70         factory = new ExcelFactory();
71         Document doc3 = factory.createDocument();
72         doc3.open();
73     }
74 }
75
```

Opening Word Document.
Opening PDF Document.
Opening Excel Document.
...Program finished with exit code 0
Press ENTER to exit console.

Exercise 3:

```

1 // Target interface
2 // class Computer {
3 //     // Required parameters
4 //     private String CPU;
5 //     private String RAM;
6 //
7 //     // Optional parameters
8 //     private String storage;
9 //     private String graphicsCard;
10
11 // Private constructor: Only Builder can access
12 private Computer(builder builder) {
13     this.CPU = builder.CPU;
14     this.RAM = builder.RAM;
15     this.storage = builder.storage;
16     this.graphicsCard = builder.graphicsCard;
17 }
18
19 public void showConfig() {
20     System.out.println("Computer Configuration:");
21     System.out.println("CPU: " + CPU);
22     System.out.println("RAM: " + RAM);
23     System.out.println("Storage: " + (storage != null ? storage : "not included"));
24     System.out.println("Graphics Card: " + (graphicsCard != null ? graphicsCard : "not included"));
25     System.out.println();
26 }
27
28 // Builder class
29 public static class Builder {
30     private String CPU;
31     private String RAM;
32     private String storage;
33     private String graphicsCard;
34
35     public Builder(String CPU, String RAM) {
36         this.CPU = CPU;
37         this.RAM = RAM;
38     }
39
40     public Builder setStorage(String storage) {
41         this.storage = storage;
42         return this;
43     }
44
45     public Builder setGraphicsCard(String graphicsCard) {
46         this.graphicsCard = graphicsCard;
47         return this;
48     }
49
50     public Computer build() {
51         return new Computer(this);
52     }
53 }
54
55 // Main class to test the Builder Pattern
56 public class Main {
57     public static void main(String[] args) {
58         // Basic computer
59         Computer basic = new Computer.Builder("Intel i7", "16GB").build();
60         basic.showConfig();
61
62         // Gaming computer
63         Computer gaming = new Computer.Builder("AMD Ryzen 7", "32GB")
64             .setStorage("1TB SSD")
65             .setGraphicsCard("NVIDIA RTX 4080")
66             .build();
67         gaming.showConfig();
68
69         // Desktop computer
70         Computer desktop = new Computer.Builder("Intel i9", "64GB")
71             .setStorage("2TB SSD")
72             .setGraphicsCard("RTX 6000")
73             .build();
74         desktop.showConfig();
75
76         // Laptop computer
77         Computer laptop = new Computer.Builder("AMD Ryzen 7", "16GB")
78             .setStorage("512GB SSD")
79             .setGraphicsCard("Intel Iris Xe")
80             .build();
81         laptop.showConfig();
82
83         // Server computer
84         Computer server = new Computer.Builder("Intel Xeon", "256GB")
85             .setStorage("10TB HDD")
86             .setGraphicsCard("NVIDIA RTX A6000")
87             .build();
88         server.showConfig();
89
90         // Main class to test the Builder Pattern
91         public class Main {
92             public static void main(String[] args) {
93                 // Basic computer
94                 Computer basic = new Computer.Builder("Intel i7", "16GB").build();
95                 basic.showConfig();
96
97                 // Gaming computer
98                 Computer gaming = new Computer.Builder("AMD Ryzen 7", "32GB")
99                     .setStorage("1TB SSD")
100                     .setGraphicsCard("NVIDIA RTX 4080")
101                     .build();
102                 gaming.showConfig();
103
104                 // Desktop computer
105                 Computer desktop = new Computer.Builder("Intel i9", "64GB")
106                     .setStorage("2TB SSD")
107                     .setGraphicsCard("RTX 6000")
108                     .build();
109                 desktop.showConfig();
110
111                 // Laptop computer
112                 Computer laptop = new Computer.Builder("AMD Ryzen 7", "16GB")
113                     .setStorage("512GB SSD")
114                     .setGraphicsCard("Intel Iris Xe")
115                     .build();
116                 laptop.showConfig();
117
118                 // Server computer
119                 Computer server = new Computer.Builder("Intel Xeon", "256GB")
120                     .setStorage("10TB HDD")
121                     .setGraphicsCard("NVIDIA RTX A6000")
122                     .build();
123                 server.showConfig();
124
125                 // Main class to test the Builder Pattern
126                 public class Main {
127                     public static void main(String[] args) {
128                         // Basic computer
129                         Computer basic = new Computer.Builder("Intel i7", "16GB").build();
130                         basic.showConfig();
131
132                         // Gaming computer
133                         Computer gaming = new Computer.Builder("AMD Ryzen 7", "32GB")
134                             .setStorage("1TB SSD")
135                             .setGraphicsCard("NVIDIA RTX 4080")
136                             .build();
137                         gaming.showConfig();
138
139                         // Desktop computer
140                         Computer desktop = new Computer.Builder("Intel i9", "64GB")
141                             .setStorage("2TB SSD")
142                             .setGraphicsCard("RTX 6000")
143                             .build();
144                         desktop.showConfig();
145
146                         // Laptop computer
147                         Computer laptop = new Computer.Builder("AMD Ryzen 7", "16GB")
148                             .setStorage("512GB SSD")
149                             .setGraphicsCard("Intel Iris Xe")
150                             .build();
151                         laptop.showConfig();
152
153                         // Server computer
154                         Computer server = new Computer.Builder("Intel Xeon", "256GB")
155                             .setStorage("10TB HDD")
156                             .setGraphicsCard("NVIDIA RTX A6000")
157                             .build();
158                         server.showConfig();
159
160                         // Main class to test the Builder Pattern
161                         public class Main {
162                             public static void main(String[] args) {
163                                 // Basic computer
164                                 Computer basic = new Computer.Builder("Intel i7", "16GB").build();
165                                 basic.showConfig();
166
167                                 // Gaming computer
168                                 Computer gaming = new Computer.Builder("AMD Ryzen 7", "32GB")
169                                     .setStorage("1TB SSD")
170                                     .setGraphicsCard("NVIDIA RTX 4080")
171                                     .build();
172                                 gaming.showConfig();
173
174                                 // Desktop computer
175                                 Computer desktop = new Computer.Builder("Intel i9", "64GB")
176                                     .setStorage("2TB SSD")
177                                     .setGraphicsCard("RTX 6000")
178                                     .build();
179                                 desktop.showConfig();
180
181                                 // Laptop computer
182                                 Computer laptop = new Computer.Builder("AMD Ryzen 7", "16GB")
183                                     .setStorage("512GB SSD")
184                                     .setGraphicsCard("Intel Iris Xe")
185                                     .build();
186                                 laptop.showConfig();
187
188                                 // Server computer
189                                 Computer server = new Computer.Builder("Intel Xeon", "256GB")
190                                     .setStorage("10TB HDD")
191                                     .setGraphicsCard("NVIDIA RTX A6000")
192                                     .build();
193                                 server.showConfig();
194
195                                 // Main class to test the Builder Pattern
196                                 public class Main {
197                                     public static void main(String[] args) {
198                                         // Basic computer
199                                         Computer basic = new Computer.Builder("Intel i7", "16GB").build();
200                                         basic.showConfig();
201
202                                         // Gaming computer
203                                         Computer gaming = new Computer.Builder("AMD Ryzen 7", "32GB")
204                                             .setStorage("1TB SSD")
205                                             .setGraphicsCard("NVIDIA RTX 4080")
206                                             .build();
207                                         gaming.showConfig();
208
209                                         // Desktop computer
210                                         Computer desktop = new Computer.Builder("Intel i9", "64GB")
211                                             .setStorage("2TB SSD")
212                                             .setGraphicsCard("RTX 6000")
213                                             .build();
214                                         desktop.showConfig();
215
216                                         // Laptop computer
217                                         Computer laptop = new Computer.Builder("AMD Ryzen 7", "16GB")
218                                             .setStorage("512GB SSD")
219                                             .setGraphicsCard("Intel Iris Xe")
220                                             .build();
221                                         laptop.showConfig();
222
223                                         // Server computer
224                                         Computer server = new Computer.Builder("Intel Xeon", "256GB")
225                                             .setStorage("10TB HDD")
226                                             .setGraphicsCard("NVIDIA RTX A6000")
227                                             .build();
228                                         server.showConfig();
229
230                                         // Main class to test the Builder Pattern
231                                         public class Main {
232                                             public static void main(String[] args) {
233                                                 // Basic computer
234                                                 Computer basic = new Computer.Builder("Intel i7", "16GB").build();
235                                                 basic.showConfig();
236
237                                                 // Gaming computer
238                                                 Computer gaming = new Computer.Builder("AMD Ryzen 7", "32GB")
239                                                     .setStorage("1TB SSD")
240                                                     .setGraphicsCard("NVIDIA RTX 4080")
241                                                     .build();
242                                                 gaming.showConfig();
243
244                                                 // Desktop computer
245                                                 Computer desktop = new Computer.Builder("Intel i9", "64GB")
246                                                     .setStorage("2TB SSD")
247                                                     .setGraphicsCard("RTX 6000")
248                                                     .build();
249                                                 desktop.showConfig();
250
251                                                 // Laptop computer
252                                                 Computer laptop = new Computer.Builder("AMD Ryzen 7", "16GB")
253                                                     .setStorage("512GB SSD")
254                                                     .setGraphicsCard("Intel Iris Xe")
255                                                     .build();
256                                                 laptop.showConfig();
257
258                                                 // Server computer
259                                                 Computer server = new Computer.Builder("Intel Xeon", "256GB")
260                                                     .setStorage("10TB HDD")
261                                                     .setGraphicsCard("NVIDIA RTX A6000")
262                                                     .build();
263                                                 server.showConfig();
264
265                                                 // Main class to test the Builder Pattern
266                                                 public class Main {
267                                                     public static void main(String[] args) {
268                                                         // Basic computer
269                                                         Computer basic = new Computer.Builder("Intel i7", "16GB").build();
270                                                         basic.showConfig();
271
272                                                         // Gaming computer
273                                                         Computer gaming = new Computer.Builder("AMD Ryzen 7", "32GB")
274                                                             .setStorage("1TB SSD")
275                                                             .setGraphicsCard("NVIDIA RTX 4080")
276                                                             .build();
277                                                         gaming.showConfig();
278
279                                                         // Desktop computer
280                                                         Computer desktop = new Computer.Builder("Intel i9", "64GB")
281                                                             .setStorage("2TB SSD")
282                                                             .setGraphicsCard("RTX 6000")
283                                                             .build();
284                                                         desktop.showConfig();
285
286                                                         // Laptop computer
287                                                         Computer laptop = new Computer.Builder("AMD Ryzen 7", "16GB")
288                                                             .setStorage("512GB SSD")
289                                                             .setGraphicsCard("Intel Iris Xe")
290                                                             .build();
291                                                         laptop.showConfig();
292
293                                                         // Server computer
294                                                         Computer server = new Computer.Builder("Intel Xeon", "256GB")
295                                                             .setStorage("10TB HDD")
296                                                             .setGraphicsCard("NVIDIA RTX A6000")
297                                                             .build();
298                                                         server.showConfig();
299
300                                                         // Main class to test the Builder Pattern
301                                                         public class Main {
302                                                             public static void main(String[] args) {
303                                                                 // Basic computer
304                                                                 Computer basic = new Computer.Builder("Intel i7", "16GB").build();
305                                                                 basic.showConfig();
306
307                                                                 // Gaming computer
308                                                                 Computer gaming = new Computer.Builder("AMD Ryzen 7", "32GB")
309                                                                    .setStorage("1TB SSD")
310                                                                    .setGraphicsCard("NVIDIA RTX 4080")
311                                                                    .build();
312                                                                 gaming.showConfig();
313
314                                                                 // Desktop computer
315                                                                 Computer desktop = new Computer.Builder("Intel i9", "64GB")
316                                                                    .setStorage("2TB SSD")
317                                                                    .setGraphicsCard("RTX 6000")
318                                                                    .build();
319                                                                 desktop.showConfig();
320
321                                                                 // Laptop computer
322                                                                 Computer laptop = new Computer.Builder("AMD Ryzen 7", "16GB")
323                                                                    .setStorage("512GB SSD")
324                                                                    .setGraphicsCard("Intel Iris Xe")
325                                                                    .build();
326                                                                 laptop.showConfig();
327
328                                                                 // Server computer
329                                                                 Computer server = new Computer.Builder("Intel Xeon", "256GB")
330                                                                    .setStorage("10TB HDD")
331                                                                    .setGraphicsCard("NVIDIA RTX A6000")
332                                                                    .build();
333                                                                 server.showConfig();
334
335                                                                 // Main class to test the Builder Pattern
336                                                                 public class Main {
337                                                                     public static void main(String[] args) {
338                                                                         // Basic computer
339                                                                         Computer basic = new Computer.Builder("Intel i7", "16GB").build();
340                                                                         basic.showConfig();
341
342                                                                         // Gaming computer
343                                                                         Computer gaming = new Computer.Builder("AMD Ryzen 7", "32GB")
344                                                                             .setStorage("1TB SSD")
345                                                                             .setGraphicsCard("NVIDIA RTX 4080")
346                                                                             .build();
347                                                                         gaming.showConfig();
348
349                                                                         // Desktop computer
350                                                                         Computer desktop = new Computer.Builder("Intel i9", "64GB")
351                                                                             .setStorage("2TB SSD")
352                                                                             .setGraphicsCard("RTX 6000")
353                                                                             .build();
354                                                                         desktop.showConfig();
355
356                                                                         // Laptop computer
357                                                                         Computer laptop = new Computer.Builder("AMD Ryzen 7", "16GB")
358                                                                             .setStorage("512GB SSD")
359                                                                             .setGraphicsCard("Intel Iris Xe")
360                                                                             .build();
361                                                                         laptop.showConfig();
362
363                                                                         // Server computer
364                                                                         Computer server = new Computer.Builder("Intel Xeon", "256GB")
365                                                                             .setStorage("10TB HDD")
366                                                                             .setGraphicsCard("NVIDIA RTX A6000")
367                                                                             .build();
368                                                                         server.showConfig();
369
370                                                                         // Main class to test the Builder Pattern
371                                                                         public class Main {
372                                                                             public static void main(String[] args) {
373                                                                                 // Basic computer
374                                                                                 Computer basic = new Computer.Builder("Intel i7", "16GB").build();
375                                                                                 basic.showConfig();
376
377                                                                                 // Gaming computer
378                                                                                 Computer gaming = new Computer.Builder("AMD Ryzen 7", "32GB")
379                                                                                     .setStorage("1TB SSD")
380                                                                                     .setGraphicsCard("NVIDIA RTX 4080")
381                                                                                     .build();
382                                                                                 gaming.showConfig();
383
384                                                                                 // Desktop computer
385                                                                                 Computer desktop = new Computer.Builder("Intel i9", "64GB")
386                                                                                     .setStorage("2TB SSD")
387                                                                                     .setGraphicsCard("RTX 6000")
388                                                                                     .build();
389                                                                                 desktop.showConfig();
390
391                                                                                 // Laptop computer
392                                                                                 Computer laptop = new Computer.Builder("AMD Ryzen 7", "16GB")
393                                                                                     .setStorage("512GB SSD")
394                                                                                     .setGraphicsCard("Intel Iris Xe")
395                                                                                     .build();
396                                                                                 laptop.showConfig();
397
398                                                                                 // Server computer
399                                                                                 Computer server = new Computer.Builder("Intel Xeon", "256GB")
400                                                                                     .setStorage("10TB HDD")
401                                                                                     .setGraphicsCard("NVIDIA RTX A6000")
402                                                                                     .build();
403                                                                                 server.showConfig();
404
405                                                                                 // Main class to test the Builder Pattern
406                                                                                 public class Main {
407                                                                                     public static void main(String[] args) {
408                                                                                         // Basic computer
409                                                                                         Computer basic = new Computer.Builder("Intel i7", "16GB").build();
410                                                                                         basic.showConfig();
411
412                                                                                         // Gaming computer
413                                                                                         Computer gaming = new Computer.Builder("AMD Ryzen 7", "32GB")
414                                                                                             .setStorage("1TB SSD")
415                                                                                             .setGraphicsCard("NVIDIA RTX 4080")
416                                                                                             .build();
417                                                                                         gaming.showConfig();
418
419                                                                                         // Desktop computer
420                                                                                         Computer desktop = new Computer.Builder("Intel i9", "64GB")
421                                                                                             .setStorage("2TB SSD")
422                                                                                             .setGraphicsCard("RTX 6000")
423                                                                                             .build();
424                                                                                         desktop.showConfig();
425
426                                                                                         // Laptop computer
427                                                                                         Computer laptop = new Computer.Builder("AMD Ryzen 7", "16GB")
428                                                                                             .setStorage("512GB SSD")
429                                                                                             .setGraphicsCard("Intel Iris Xe")
430                                                                                             .build();
431                                                                                         laptop.showConfig();
432
433                                                                                         // Server computer
434                                                                                         Computer server = new Computer.Builder("Intel Xeon", "256GB")
435                                                                                             .setStorage("10TB HDD")
436                                                                                             .setGraphicsCard("NVIDIA RTX A6000")
437                                                                                             .build();
438                                                                                         server.showConfig();
439
440                                                                                         // Main class to test the Builder Pattern
441                                                                                         public class Main {
442                                                                                             public static void main(String[] args) {
443                                                                                                 // Basic computer
444                                                                                                 Computer basic = new Computer.Builder("Intel i7", "16GB").build();
445                                                                                                 basic.showConfig();
446
447                                                                                                 // Gaming computer
448                                                                                                 Computer gaming = new Computer.Builder("AMD Ryzen 7", "32GB")
449                                                                                                     .setStorage("1TB SSD")
450                                                                                                     .setGraphicsCard("NVIDIA RTX 4080")
451                                                                                                     .build();
452                                                                                                 gaming.showConfig();
453
454                                                                                                 // Desktop computer
455                                                                                                 Computer desktop = new Computer.Builder("Intel i9", "64GB")
456                                                                                                     .setStorage("2TB SSD")
457                                                                                                     .setGraphicsCard("RTX 6000")
458                                                                                                     .build();
459                                                                                                 desktop.showConfig();
460
461                                                                                                 // Laptop computer
462                                                                                                 Computer laptop = new Computer.Builder("AMD Ryzen 7", "16GB")
463                                                                                                     .setStorage("512GB SSD")
464                                                                                                     .setGraphicsCard("Intel Iris Xe")
465                                                                                                     .build();
466                                                                                                 laptop.showConfig();
467
468                                                                                                 // Server computer
469                                                                                                 Computer server = new Computer.Builder("Intel Xeon", "256GB")
470                                                                                                     .setStorage("10TB HDD")
471                                                                                                     .setGraphicsCard("NVIDIA RTX A6000")
472                                                                                                     .build();
473                                                                                                 server.showConfig();
474
475                                                                                                 // Main class to test the Builder Pattern
476                                                                                                 public class Main {
477                                                                                                     public static void main(String[] args) {
478                                                                                                         // Basic computer
479                                                                                                         Computer basic = new Computer.Builder("Intel i7", "16GB").build();
480                                                                                                         basic.showConfig();
481
482                                                                                                         // Gaming computer
483                                                                                                         Computer gaming = new Computer.Builder("AMD Ryzen 7", "32GB")
484                                                                                                             .setStorage("1TB SSD")
485                                                                                                             .setGraphicsCard("NVIDIA RTX 4080")
486                                                                                                             .build();
487                                                                                                         gaming.showConfig();
488
489                                                                                                         // Desktop computer
490                                                                                                         Computer desktop = new Computer.Builder("Intel i9", "64GB")
491                                                                                                             .setStorage("2TB SSD")
492                                                                                                             .setGraphicsCard("RTX 6000")
493                                                                                                             .build();
494                                                                                                         desktop.showConfig();
495
496                                                                                                         // Laptop computer
497                                                                                                         Computer laptop = new Computer.Builder("AMD Ryzen 7", "16GB")
498                                                                                                             .setStorage("512GB SSD")
499                                                                                                             .setGraphicsCard("Intel Iris Xe")
500                                                                                                             .build();
501                                                                                                         laptop.showConfig();
502
499                                     }
500                                 }
501                             }
502                         }
503                     }
504                 }
505             }
506         }
507     }
508 }

```

Exercise 4:

```

8
9 // Target interface
10 interface PaymentProcessor {
11     void processPayment(double amount);
12 }
13
14 // Adaptee classes (3rd party APIs with different method names)
15 class PayPalGateway {
16     public void sendPayment(double amount) {
17         System.out.println("Processing PayPal payment of $" + amount);
18     }
19 }
20
21 class StripeGateway {
22     public void makePayment(double amount) {
23         System.out.println("Processing Stripe payment of $" + amount);
24     }
25 }
26
27 // Adapter classes
28 class PayPalAdapter implements PaymentProcessor {
29     private PayPalGateway paypal = new PayPalGateway();
30
31     public void processPayment(double amount) {
32         paypal.sendPayment(amount);
33     }
34 }
35
36 class StripeAdapter implements PaymentProcessor {
37     private StripeGateway stripe = new StripeGateway();
38
39     public void processPayment(double amount) {
40         stripe.makePayment(amount);
41     }
42 }
43
44 // Main class to test Adapter Pattern

```

```

Processing PayPal payment of $150.75
Processing Stripe payment of $320.0

...Program finished with exit code 0
Press ENTER to exit console.

```

Exercise 5:

```

34 // Concrete decorators
35 class SMSNotifierDecorator extends NotifierDecorator {
36     public SMSNotifierDecorator(Notifier notifier) {
37         super(notifier);
38     }
39
40     public void send(String message) {
41         super.send(message);
42         System.out.println("Sending SMS: " + message);
43     }
44 }
45
46 class SlackNotifierDecorator extends NotifierDecorator {
47     public SlackNotifierDecorator(Notifier notifier) {
48         super(notifier);
49     }
50
51     public void send(String message) {
52         super.send(message);
53         System.out.println("Sending Slack: " + message);
54     }
55 }
56
57 // Main class to test Decorator Pattern
58 public class Main {
59     public static void main(String[] args) {
60         // Basic email notification
61         Notifier notifier = new EmailNotifier();
62
63         // Add SMS
64         notifier = new SMSNotifierDecorator(notifier);
65
66         // Add Slack
67         notifier = new SlackNotifierDecorator(notifier);
68
69         // Send notification
70         notifier.send("System Alert: Server down!");
71     }
72 }
73

```

Sending Email: System Alert: Server down!
 Sending SMS: System Alert: Server down!
 Sending Slack: System Alert: Server down!
 ...Program finished with exit code 0
 Press ENTER to exit console.

Exercise 6:

```

23 private void loadFromDisk() {
24     System.out.println("Loading image: " + filename);
25 }
26
27 public void display() {
28     System.out.println("Displaying image: " + filename);
29 }
30 }
31
32 // Proxy object
33 class ProxyImage implements Image {
34     private RealImage realImage;
35     private String filename;
36
37     public ProxyImage(String filename) {
38         this.filename = filename;
39     }
40
41     public void display() {
42         if (realImage == null) {
43             realImage = new RealImage(filename); // lazy loading
44         }
45         realImage.display(); // caching
46     }
47 }
48
49 // Main class to test Proxy Pattern
50 public class Main {
51     public static void main(String[] args) {
52         Image image1 = new ProxyImage("photo1.jpg");
53         Image image2 = new ProxyImage("photo2.jpg");
54
55         // First time: Loads and displays
56         image1.display();
57
58         // Second time: Uses cached image
59         image1.display();
60
61         // Loads and displays new image
62         image2.display();
63     }
64 }
65

```

Loading image: photo1.jpg
 Displaying image: photo1.jpg
 Displaying image: photo1.jpg
 Loading image: photo2.jpg
 Displaying image: photo2.jpg
 ...Program finished with exit code 0
 Press ENTER to exit console.

Exercise 7:

```

35- public void notifyObservers(String stockName, double price) {
36-     for (Observer o : observers) {
37-         o.update(stockName, price);
38-     }
39- }
40-
41- public void updateStockPrice(String stockName, double price) {
42-     System.out.println("Stock Update: " + stockName + " is now $" + price);
43-     notifyObservers(stockName, price);
44- }
45- }
46-
47- // Concrete Observers
48- class MobileApp implements Observer {
49-     public void update(String stockName, double price) {
50-         System.out.println("📱 Mobile App - " + stockName + " price updated to $" + price);
51-     }
52- }
53-
54- class WebApp implements Observer {
55-     public void update(String stockName, double price) {
56-         System.out.println("🌐 Web App - " + stockName + " price updated to $" + price);
57-     }
58- }
59-
60- // Main class to test Observer Pattern
61- public class Main {
62-     public static void main(String[] args) {
63-         StockMarket stockMarket = new StockMarket();
64-
65-         Observer mobileApp = new MobileApp();
66-         Observer webApp = new WebApp();
67-
68-         stockMarket.registerObserver(mobileApp);
69-         stockMarket.registerObserver(webApp);
70-
71-         stockMarket.updateStockPrice("AAPL", 180.25);
72-         stockMarket.updateStockPrice("GOOGL", 2790.75);
73-
74-         stockMarket.removeObserver(webApp);
75-         stockMarket.updateStockPrice("TSLA", 950.00);
76-     }
77- }
78-

```

```

Stock Update: AAPL is now $180.25
📱 Mobile App - AAPL price updated to $180.25
🌐 Web App - AAPL price updated to $180.25
Stock Update: GOOGL is now $2790.75
📱 Mobile App - GOOGL price updated to $2790.75
🌐 Web App - GOOGL price updated to $2790.75
Stock Update: TSLA is now $950.0
📱 Mobile App - TSLA price updated to $950.0

```

```

..Program finished with exit code 0
Press ENTER to exit console.

```

Exercise 8:

```

15 class CreditCardPayment implements PaymentStrategy {
16     public void pay(double amount) {
17         System.out.println("Paid $" + amount + " using Credit Card.");
18     }
19 }
20
21 class PayPalPayment implements PaymentStrategy {
22     public void pay(double amount) {
23         System.out.println("Paid $" + amount + " using PayPal.");
24     }
25 }
26
27 // Context class
28 class PaymentContext {
29     private PaymentStrategy paymentStrategy;
30
31     public void setPaymentStrategy(PaymentStrategy strategy) {
32         this.paymentStrategy = strategy;
33     }
34
35     public void payAmount(double amount) {
36         if (paymentStrategy == null) {
37             System.out.println("No payment strategy selected.");
38         } else {
39             paymentStrategy.pay(amount);
40         }
41     }
42 }
43
44 // Main class to test Strategy Pattern
45 public class Main {
46     public static void main(String[] args) {
47         PaymentContext context = new PaymentContext();
48
49         // Pay using Credit Card
50         context.setPaymentStrategy(new CreditCardPayment());
51         context.payAmount(250.00);
52
53         // Pay using PayPal
54         context.setPaymentStrategy(new PayPalPayment());
55         context.payAmount(100.00);
56     }
57 }
58

```

```

Paid $250.0 using Credit Card.
Paid $100.0 using PayPal.

...Program finished with exit code 0
Press ENTER to exit console.

```

Exercise 9:

```

41 public LightOnCommand(Light light) {
42     this.light = light;
43 }
44
45 public void execute() {
46     light.turnOff();
47 }
48 }
49
50 // Invoker class
51 class RemoteControl {
52     private Command command;
53
54     public void setCommand(Command command) {
55         this.command = command;
56     }
57
58     public void pressButton() {
59         if (command != null) {
60             command.execute();
61         } else {
62             System.out.println("No command assigned.");
63         }
64     }
65 }
66
67 // Main class to test Command Pattern
68 public class Main {
69     public static void main(String[] args) {
70         Light livingRoomLight = new Light();
71
72         Command lightOn = new LightOnCommand(livingRoomLight);
73         Command lightOff = new LightOffCommand(livingRoomLight);
74
75         RemoteControl remote = new RemoteControl();
76
77         remote.setCommand(lightOn);
78         remote.pressButton(); // Turn on
79
80         remote.setCommand(lightOff);
81         remote.pressButton(); // Turn off
82     }
83 }
84

```

```

Light is ON
Light is OFF

...Program finished with exit code 0
Press ENTER to exit console.

```


Exercise 10:

```

45
46- public StudentController(Student model, StudentView view) {
47-     this.model = model;
48-     this.view = view;
49- }
50
51- public void setStudentName(String name) {
52-     model.setName(name);
53- }
54
55- public void setStudentGrade(String grade) {
56-     model.setGrade(grade);
57- }
58
59- public void updateView() {
60-     view.displayStudentDetails(model.getName(), model.getId(), model.getGrade());
61- }
62- }
63
64 // Main class to test MVC Pattern
65- public class Main {
66-     public static void main(String[] args) {
67-         // Create model
68-         Student student = new Student("Riya", "S102", "A");
69-
70-         // Create view
71-         StudentView view = new StudentView();
72-
73-         // Create controller
74-         StudentController controller = new StudentController(student, view);
75-
76-         // Initial display
77-         controller.updateView();
78-
79-         // Update model via controller
80-         controller.setStudentName("Riya Sen");
81-         controller.setStudentGrade("A+");
82-
83-         // Updated display
84-         controller.updateView();
85-     }
86- }
87

```

input

```

Student Info:
Name: Riya
ID: S102
Grade: A

Student Info:
Name: Riya Sen
ID: S102
Grade: A+

```

Exercise 11:

```

9 // Repository interface
10- interface CustomerRepository {
11-     String findCustomerById(String customerId);
12- }
13
14 // Concrete repository implementation
15- class CustomerRepositoryImpl implements CustomerRepository {
16-     public String findCustomerById(String customerId) {
17-         // Simulate fetching from database
18-         return "Customer[id=" + customerId + ", name=John Doe]";
19-     }
20- }
21
22 // Service class that depends on repository
23- class CustomerService {
24-     private CustomerRepository repository;
25-
26-     // Constructor injection
27-     public CustomerService(CustomerRepository repository) {
28-         this.repository = repository;
29-     }
30-
31-     public void displayCustomer(String customerId) {
32-         String customerData = repository.findCustomerById(customerId);
33-         System.out.println(" Retrieved: " + customerData);
34-     }
35- }
36
37 // Main class to test Dependency Injection
38- public class Main {
39-     public static void main(String[] args) {
40-         // Inject repository implementation into service
41-         CustomerRepository repo = new CustomerRepositoryImpl();
42-         CustomerService service = new CustomerService(repo);
43-
44-         // Use the service
45-         service.displayCustomer("C123");
46-     }
47- }
48

```

```

Retrieved: Customer[id=C123, name=John Doe]

..Program finished with exit code 0
Press ENTER to exit console.

```