

```
1: /*
2:  * Car.java
3:  * Dan Armendariz
4:  * Computer Science E-76
5:  *
6:  * A brief introduction to methods.
7:  *
8:  */
9:
10: class Car {
11:
12:     private String make;
13:     private String model;
14:     private int year;
15:     private double speed;
16:     private double maxSpeed;
17:
18:     public Car(String mk, String mdl, int yr) {
19:         make = mk;
20:         model = mdl;
21:         year = yr;
22:     }
23:
24:     public void setMaxSpeed(double max) {
25:         maxSpeed = max;
26:     }
27:
28:     public boolean setSpeed(double spd) {
29:         if(spd > maxSpeed) return false;
30:
31:         speed = spd;
32:         return true;
33:     }
34:
35:     public double getSpeed() {
36:         return speed;
37:     }
38:
39:     public String getMake() {
40:         return make;
41:     }
42:
43:     public String getModel() {
44:         return model;
45:     }
```

```
46:
47:     public int getYear() {
48:         return year;
49:     }
50:
51: }
```

```
1: /*
2:  * Code10.java
3:  * Dan Armendariz
4:  * Computer Science E-76
5:  *
6:  * String input and comparison - fixed!
7:  *
8:  */
9:
10: // allow us use of the keyboard scanner. More information from the docs:
11: // http://java.sun.com/javase/6/docs/api/java/util/Scanner.html
12: import java.util.Scanner;
13:
14: class Code10 {
15:
16:     public static void main(String [] args) {
17:
18:         String str = "Hello, world";
19:         String input = null;
20:
21:         // instantiate the Scanner class, accessing data from the keyboard
22:         Scanner keyboard = new Scanner(System.in);
23:
24:         System.out.print("Please type a string: ");
25:
26:         // wait for the user to enter an integer
27:         try {
28:             input = keyboard.nextLine();
29:         } catch(Exception e) {
30:             System.out.println("Invalid input! Quitting..");
31:             System.exit(1);
32:         }
33:
34:         // test to see if what the user entered matches our string, using
35:         // the equals method.
36:         if(str.equals(input)) {
37:             System.out.println("Strings match! :-)");
38:         } else {
39:             System.out.println("Strings do not match! :-(");
40:         }
41:     }
42: }
43:
44: }
```

```
1: /*
2:  * Code11.java
3:  * Dan Armendariz
4:  * Computer Science E-76
5:  *
6:  * Introduction to arrays.
7:  *
8:  */
9:
10: class Code11 {
11:
12:     public static void main(String [] args) {
13:
14:         // declare the array
15:         int[] grades;
16:
17:         // allocate memory for 5 indices
18:         grades = new int[5];
19:
20:         // assign some values to the array
21:         grades[0] = 100;
22:         grades[1] = 76;
23:         grades[2] = 92;
24:         grades[3] = 95;
25:         grades[4] = 14;
26:
27:         // print out each value
28:         System.out.println(grades[0]);
29:         System.out.println(grades[1]);
30:         System.out.println(grades[2]);
31:         System.out.println(grades[3]);
32:         System.out.println(grades[4]);
33:
34:     }
35:
36: }
```

```
1: /*
2:  * Code12.java
3:  * Dan Armendariz
4:  * Computer Science E-76
5:  *
6:  * A less stupid way of printing an array's contents.
7:  *
8:  */
9:
10: class Code12 {
11:
12:     public static void main(String [] args) {
13:
14:         // declare the array
15:         int[] grades;
16:
17:         // allocate memory for 5 indices
18:         grades = new int[5];
19:
20:         // assign some values to the array
21:         grades[0] = 100;
22:         grades[1] = 76;
23:         grades[2] = 92;
24:         grades[3] = 95;
25:         grades[4] = 14;
26:
27:         for(int i = 0; i < grades.length; i++) {
28:             System.out.println("Grade "+(i+1)+" : "+grades[i]);
29:         }
30:
31:     }
32:
33: }
```

```
1: /*
2:  * Code13.java
3:  * Dan Armendariz
4:  * Computer Science E-76
5:  *
6:  * A less stupid way of printing an array's contents, and an optimization.
7:  *
8:  */
9:
10: class Code13 {
11:
12:     public static void main(String [] args) {
13:
14:         // declare the array
15:         int[] grades;
16:
17:         // allocate memory for 5 indices
18:         grades = new int[5];
19:
20:         // assign some values to the array
21:         grades[0] = 100;
22:         grades[1] = 76;
23:         grades[2] = 92;
24:         grades[3] = 95;
25:         grades[4] = 14;
26:
27:         // pre-store the length into a variable
28:         for(int i = 0, j = grades.length; i < j; i++) {
29:             System.out.println("Grade "+(i+1)+" : "+grades[i]);
30:         }
31:
32:     }
33:
34: }
```

```
1: /*
2:  * Code14.java
3:  * Dan Armendariz
4:  * Computer Science E-76
5:  *
6:  * Printing an array's contents with a while loop.
7:  *
8:  */
9:
10: class Code14 {
11:
12:     public static void main(String [] args) {
13:
14:         // declare the array
15:         int[] grades;
16:
17:         // allocate memory for 5 indices
18:         grades = new int[5];
19:
20:         // assign some values to the array
21:         grades[0] = 100;
22:         grades[1] = 76;
23:         grades[2] = 92;
24:         grades[3] = 95;
25:         grades[4] = 14;
26:
27:         int i = 0, j = grades.length;
28:
29:         // loop the code while the condition evaluates to true.
30:         while(i < j) {
31:
32:             System.out.println("Grade "+(i+1)+": "+grades[i]);
33:             i++;
34:
35:         }
36:
37:     }
38:
39: }
```

```
1: /*
2:  * Code15.java
3:  * Dan Armendariz
4:  * Computer Science E-76
5:  *
6:  * Printing an array's contents with a do-while loop.
7:  *
8:  */
9:
10: class Code15 {
11:
12:     public static void main(String [] args) {
13:
14:         // declare the array
15:         int[] grades;
16:
17:         // allocate memory for 5 indices
18:         grades = new int[5];
19:
20:         // assign some values to the array
21:         grades[0] = 100;
22:         grades[1] = 76;
23:         grades[2] = 92;
24:         grades[3] = 95;
25:         grades[4] = 14;
26:
27:         int i = 0, j = grades.length;
28:
29:         // evaluate a condition after an initial run. In other words,
30:         // this loop is guaranteed to run at least once.
31:         do {
32:             System.out.println("Grade "+(i+1)+": "+grades[i]);
33:             } while(++i < j);
34:
35:     }
36:
37: }
```

```
1: /*
2:  * Code16.java
3:  * Dan Armendariz
4:  * Computer Science E-76
5:  *
6:  * Using a do-while loop to guarantee valid input from the user.
7:  *
8:  */
9:
10: import java.util.Scanner;
11:
12: class Code16 {
13:
14:     public static void main(String [] args) {
15:
16:         boolean invalid;
17:         int input = 0;
18:
19:         // instantiate the Scanner class, accessing data from the keyboard
20:         Scanner keyboard = new Scanner(System.in);
21:
22:         do {
23:
24:             invalid = false;
25:             System.out.print("Please enter an integer: ");
26:
27:             // try inputting an integer, if a user doesn't ..
28:             try {
29:
30:                 input = keyboard.nextInt();
31:
32:             } catch(Exception e) {
33:                 // an exception will be thrown, and we can catch it to alert
34:                 // the user that something bad happened.
35:
36:                 System.out.println("Invalid input! Please try again..");
37:                 invalid = true;
38:                 keyboard.next();
39:             }
40:
41:         } while (invalid);
42:
43:         System.out.println("You have finally entered a valid integer: "+input);
44:
45:     }
```

```
46:
47: }
```

```
1: /*
2:  * Code17.java
3:  * Dan Armendariz
4:  * Computer Science E-76
5:  *
6:  * A brief introduction to methods.
7:  *
8:  */
9:
10: class Code17 {
11:
12:     // declare a field
13:     private int num;
14:
15:     public static void main(String [] args) {
16:         Code17 myObj = new Code17();
17:         System.out.println("Val: " + myObj.get());
18:         myObj.set(2);
19:         System.out.println("Val: " + myObj.get());
20:     }
21:
22:     // declare a Constructor for the class and initialize our fields.
23:     public Code17() {
24:         num = 0;
25:     }
26:
27:     // define a public 'set' function to modify state
28:     public void set(int val) {
29:         num = val;
30:     }
31:
32:     // define a public 'get' function to get returned the current state
33:     public int get() {
34:         return num;
35:     }
36:
37: }
```

```
1: /*
2:  * Code18.java
3:  * Dan Armendariz
4:  * Computer Science E-76
5:  *
6:  * A brief introduction to methods.
7:  *
8:  */
9:
10: class Code18 {
11:
12:     private static void printSuccess(Car myCar) {
13:         System.out.println("Wow, your "+myCar.getYear()+" "+myCar.getMake()+
14:             " "+myCar.getModel()+" is really booking it at "+
15:             myCar.getSpeed()+" mph!");
16:     }
17:
18:
19:     private static void printFailure(Car myCar) {
20:         System.out.println("Too bad, your "+myCar.getYear()+" "+myCar.getMake()+
21:             " "+myCar.getModel()+" can't go that fast.");
22:     }
23:
24:     public static void main(String [] args) {
25:         Car tortoise = new Car("Toyota", "Camry", 2009);
26:         Car hare = new Car("Ferrari", "F430", 2009);
27:
28:         tortoise.setMaxSpeed(100.0);
29:         hare.setMaxSpeed(200.0);
30:
31:         if(hare.setSpeed(155.0))
32:             printSuccess(hare);
33:         else
34:             printFailure(hare);
35:
36:         if(tortoise.setSpeed(135.0))
37:             printSuccess(tortoise);
38:         else
39:             printFailure(tortoise);
40:     }
41:
42:
43: }
```

```
1: /*
2:  * Code19.java
3:  * Dan Armendariz
4:  * Computer Science E-76
5:  *
6:  * Demonstrates implications of passing data into methods by value.
7:  *
8:  */
9:
10: class Code19 {
11:
12:     // define two integer fields
13:     private static int firstNum;
14:     private static int secondNum;
15:
16:     // an object that contains two pieces of data: an int x and an int y
17:     // The constructor for this class accepts two values that are then
18:     // inserted into these fields.
19:     private static class Point {
20:         public int x;
21:         public int y;
22:
23:         public Point(int first, int second) {
24:             x = first;
25:             y = second;
26:         }
27:     }
28:
29:     // Accepts two integers as parameters and should swap them.
30:     private static void swap(int x, int y) {
31:         int temp;
32:
33:         temp = x;
34:         x = y;
35:         y = temp;
36:     }
37:
38:     // Accepts a Point object and swaps the data in the x and y fields.
39:     private static void swap(Point a) {
40:         int temp;
41:
42:         temp = a.x;
43:         a.x = a.y;
44:         a.y = temp;
45:     }
}
```

```
46:
47:     public static void main(String [] args) {
48:
49:         // assign some values to our integer fields
50:         firstNum = 1;
51:         secondNum = 2;
52:
53:         // instantiate a new Point object, and provide some data
54:         Point a = new Point(3, 4);
55:
56:         // attempt to swap the data in the two integer fields
57:         System.out.println("firstNum: "+firstNum+", secondNum: "+secondNum);
58:         swap(firstNum, secondNum);
59:         System.out.println("firstNum: "+firstNum+", secondNum: "+secondNum);
60:
61:         // attempt to swap the data in the Point object
62:         System.out.println("a.x: "+a.x+", a.y: "+a.y);
63:         swap(a);
64:         System.out.println("a.x: "+a.x+", a.y: "+a.y);
65:
66:     }
67:
68: }
```

```
1: /*
2:  * Code1.java
3:  * Dan Armendariz
4:  * Computer Science E-76
5:  *
6:  * A very basic Hello, World application.
7:  *
8:  */
9:
10: class Code1 {
11:
12:     public static void main(String [] args) {
13:         System.out.print("Hello, World!");
14:         System.out.println();
15:     }
16:
17: }
```

```
1: /*
2:  * Code20.java
3:  * Dan Armendariz
4:  * Computer Science E-76
5:  *
6:  * Demonstrates class inheritance.
7:  *
8:  */
9:
10: class Code20 {
11:
12:     private static Computer macPro;
13:     private static Laptop macBookAir;
14:     private static Server xserve;
15:
16:     private static void showRunningMachines() {
17:         System.out.println();
18:
19:         System.out.println("The following machines are turned on:");
20:
21:         if(macPro.isOn()) System.out.println("- "+macPro.model);
22:         if(macBookAir.isOn()) System.out.print("- "+macBookAir.model);
23:         if(xserve.isOn()) System.out.println("- "+xserve.model);
24:
25:         System.out.println();
26:
27:     }
28:
29:     public static void main(String [] args) {
30:
31:         macPro = new Computer("Apple", "MacPro", 2009, 2.8);
32:         macBookAir = new Laptop("Apple", "MacBook Air", 2009, 1.8);
33:         xserve = new Server("Apple", "Xserve", 2010, 3.2);
34:
35:         macPro.turnOn();
36:         macBookAir.turnOn();
37:
38:         macBookAir.setBattery(0.5);
39:         xserve.setRackHeight(1);
40:
41:         System.out.println(macPro.built + " " + macPro.make + " " + macPro.model
42:             + " runs at " + macPro.speed + "ghz.");
43:
44:         System.out.println(macBookAir.built + " " + macBookAir.make + " " +
45:             macBookAir.model + " runs at " + macBookAir.speed +
```



```
46:                                     "ghz and has a battery level of "+
47:                                     (macBookAir.getBattery() * 100) + "%.");
48:
49:     System.out.println(xserve.built + " " + xserve.make + " " + xserve.model +
50:                         " runs at " + xserve.speed + "ghz and is " +
51:                         xserve.getRackHeight() + "U tall.");
52:
53:     showRunningMachines();
54:
55:     System.out.println("\nUh oh, MacBook Air's battery is dying..");
56:     macBookAir.setBattery(0.0);
57:
58:     showRunningMachines();
59:
60: }
61:
62: }
```

```
1: /*
2:  * Code2.java
3:  * Dan Armendariz
4:  * Computer Science E-76
5:  *
6:  * Defining and assigning values to fields and local variables.
7:  *
8:  */
9:
10: class Code2 {
11:
12:     // define a field
13:     private static int num;
14:
15:     public static void main(String [] args) {
16:         // local variable
17:         int anotherNum = 3;
18:
19:         // assigning a value to a variable
20:         num = 2;
21:
22:         // print out some information
23:         System.out.println("num: " + num);
24:         System.out.println("anotherNum: " + anotherNum);
25:     }
26:
27: }
```

```
1: /*
2:  * Code3.java
3:  * Dan Armendariz
4:  * Computer Science E-76
5:  *
6:  * Assigning values to various types of variables.
7:  *
8:  */
9:
10: class Code3 {
11:
12:     public static void main(String [] args) {
13:
14:         int number = 123;
15:         String str = "Hello, World!";
16:
17:         // print out an integer
18:         System.out.println("number: "+number);
19:
20:         // print out a string
21:         System.out.println("str: "+str);
22:
23:         // print out a substring, a method of the String class
24:         // a list of methods is available from:
25:         // http://java.sun.com/javase/6/docs/api/java/lang/String.html
26:         System.out.println("substring: "+str.substring(7, 12));
27:
28:     }
29:
30: }
```

```
1: /*
2:  * Code4.java
3:  * Dan Armendariz
4:  * Computer Science E-76
5:  *
6:  * Changing variable type through typecasting.
7:  *
8:  */
9:
10: class Code4 {
11:
12:     public static void main(String [] args) {
13:
14:         int myInt = 123;
15:         double myDouble = 123.0;
16:         String myStr = "123";
17:         String txtStr = "Hello, World!";
18:
19:         // Integer math, is there a problem?
20:         System.out.println("myInt/5: " + (myInt / 5));
21:
22:         // modulus
23:         System.out.println("myInt%5: " + (myInt % 5));
24:
25:         // Double math, that's better
26:         System.out.println("myDouble/5: " + (myDouble / 5));
27:
28:         // Double math by type casting an int
29:         System.out.println("(double)myInt/5: "+( (double)myInt / 5));
30:
31:         // Another type-caste by forcing double math
32:         System.out.println("myInt/5.0: "+(myInt / 5.0));
33:
34:         // Attempt to convert a string to int
35:         //System.out.println("Converting myStr to int: "+ (int)myStr);
36:
37:         // Convert a string to an int
38:         System.out.println("Converting myStr to int: "+Integer.parseInt(myStr));
39:
40:         // Convert another string to an int
41:         //System.out.println("Converting txtStr to int: "+Integer.parseInt(txtStr));
42:
43:     }
44: }
```

```
1: /*
2:  * Code5.java
3:  * Dan Armendariz
4:  * Computer Science E-76
5:  *
6:  * Reading values from the keyboard, if statements, boolean expressions.
7:  *
8:  */
9:
10: // allow us use of the keyboard scanner. More information from the docs:
11: // http://java.sun.com/javase/6/docs/api/java/util/Scanner.html
12: import java.util.Scanner;
13:
14: class Code5 {
15:
16:     public static void main(String [] args) {
17:
18:         int number = 123;
19:
20:         // instantiate the Scanner class, accessing data from the keyboard
21:         Scanner keyboard = new Scanner(System.in);
22:
23:         // wait for the user to enter an integer
24:         int input = keyboard.nextInt();
25:
26:         // test to see if what the user entered matches our number.
27:         if(input == number) {
28:             System.out.println("Numbers match! :-");
29:         } else {
30:             System.out.println("Numbers do not match! :-(");
31:         }
32:     }
33: }
34:
35: }
```

```
1: /*
2:  * Code6.java
3:  * Dan Armendariz
4:  * Computer Science E-76
5:  *
6:  * A better version of reading values from the keyboard.
7:  *
8:  */
9:
10: // allow us use of the keyboard scanner. More information from the docs:
11: // http://java.sun.com/javase/6/docs/api/java/util/Scanner.html
12: import java.util.Scanner;
13:
14: class Code6 {
15:
16:     public static void main(String [] args) {
17:
18:         int number = 123;
19:
20:         // instantiate the Scanner class, accessing data from the keyboard
21:         Scanner keyboard = new Scanner(System.in);
22:
23:         System.out.print("Please enter an integer: ");
24:
25:         // wait for the user to enter an integer
26:         int input = keyboard.nextInt();
27:
28:         // test to see if what the user entered matches our number.
29:         if(input == number) {
30:             System.out.println("Numbers match! :-");
31:         } else {
32:             System.out.println("Numbers do not match! :-(");
33:         }
34:     }
35: }
36:
37: }
```

```
1: /*
2:  * Code7.java
3:  * Dan Armendariz
4:  * Computer Science E-76
5:  *
6:  * A better version of reading values from the keyboard, with exception
7:  * handling.
8:  *
9:  */
10:
11: // allow us use of the keyboard scanner. More information from the docs:
12: // http://java.sun.com/javase/6/docs/api/java/util/Scanner.html
13: import java.util.Scanner;
14:
15: class Code7 {
16:
17:     public static void main(String [] args) {
18:
19:         int number = 123;
20:         int input = 0;
21:
22:         // instantiate the Scanner class, accessing data from the keyboard
23:         Scanner keyboard = new Scanner(System.in);
24:
25:         System.out.print("Please enter an integer: ");
26:
27:         // try inputting an integer, if a user doesn't ..
28:         try {
29:
30:             input = keyboard.nextInt();
31:
32:         } catch(Exception e) {
33:             // an exception will be thrown, and we can catch it to alert
34:             // the user that something bad happened.
35:
36:             System.out.println("Invalid input! Quitting..");
37:             System.exit(1);
38:         }
39:
40:         // test to see if what the user entered matches our number.
41:         if(input == number) {
42:             System.out.println("Numbers match! :-)");
43:         } else {
44:             System.out.println("Numbers do not match! :-(");
45:         }

```

```
46:
47:     }
48:
49: }
```

```
1: /*
2:  * Code8.java
3:  * Dan Armendariz
4:  * Computer Science E-76
5:  *
6:  * Comparing input via a switch.
7:  *
8:  */
9:
10: // allow us use of the keyboard scanner. More information from the docs:
11: // http://java.sun.com/javase/6/docs/api/java/util/Scanner.html
12: import java.util.Scanner;
13:
14: class Code8 {
15:
16:     public static void main(String [] args) {
17:
18:         int input = 0;
19:
20:         // instantiate the Scanner class, accessing data from the keyboard
21:         Scanner keyboard = new Scanner(System.in);
22:
23:         System.out.print("Please enter an integer: ");
24:
25:         // try inputting an integer, if a user doesn't ..
26:         try {
27:
28:             input = keyboard.nextInt();
29:
30:         } catch(Exception e) {
31:             // an exception will be thrown, and we can catch it to alert
32:             // the user that something bad happened.
33:
34:             System.out.println("Invalid input! Quitting..");
35:             System.exit(1);
36:         }
37:
38:
39:         // determine which number the user selected
40:         switch(input) {
41:             case 1: System.out.println("You're number one!"); break;
42:             case 3: System.out.println("Third time's a charm!"); break;
43:             case 6: System.out.println("That matches my file name!"); break;
44:             default: System.out.println("That's a boring number.");
45:         }
```

```
46:
47:     }
48:
49: }
```

```
1: /*
2:  * Code9.java
3:  * Dan Armendariz
4:  * Computer Science E-76
5:  *
6:  * String input and comparison
7:  *
8:  */
9:
10: // allow us use of the keyboard scanner. More information from the docs:
11: // http://java.sun.com/javase/6/docs/api/java/util/Scanner.html
12: import java.util.Scanner;
13:
14: class Code9 {
15:
16:     public static void main(String [] args) {
17:
18:         String str = "Hello, world";
19:         String input = null;
20:
21:         // instantiate the Scanner class, accessing data from the keyboard
22:         Scanner keyboard = new Scanner(System.in);
23:
24:         System.out.print("Please type a string: ");
25:
26:         // wait for the user to enter an integer
27:         try {
28:             input = keyboard.nextLine();
29:         } catch(Exception e) {
30:             System.out.println("Invalid input! Quitting..");
31:             System.exit(1);
32:         }
33:
34:         // test to see if what the user entered matches our string.
35:         if(str == input) {
36:             System.out.println("Strings match! :-)");
37:         } else {
38:             System.out.println("Strings do not match! :-(");
39:         }
40:
41:     }
42:
43: }
```

```
1: /*
2:  * Computer.java
3:  * Dan Armendariz
4:  * Computer Science E-76
5:  *
6:  * Demonstrates class inheritance; a parent class for 'computer' sub-classes.
7:  *
8:  */
9:
10: class Computer {
11:
12:     public String make;
13:     public String model;
14:     public int built;
15:     public double speed;
16:     private boolean on;
17:
18:     public Computer(String mk, String mdl, int yr, double spd) {
19:         make = mk;
20:         model = mdl;
21:         built = yr;
22:         speed = spd;
23:         on = false;
24:     }
25:
26:     public void turnOn() {
27:         on = true;
28:     }
29:
30:     public void turnOff() {
31:         on = false;
32:     }
33:
34:     public boolean isOn() {
35:         return on;
36:     }
37:
38: }
```

```
1: /*
2:  * Laptop.java
3:  * Dan Armendariz
4:  * Computer Science E-76
5:  *
6:  * Demonstrates class inheritance; Laptop is a subclass of Computer.
7:  *
8:  */
9:
10: class Laptop extends Computer {
11:
12:     private double battery;
13:
14:     public Laptop(String mk, String mdl, int yr, double spd) {
15:         super(mk, mdl, yr, spd);
16:     }
17:
18:     public void setBattery(double lvl) {
19:         battery = lvl;
20:
21:         if (battery < 0.01) super.turnOff();
22:     }
23:
24:     public double getBattery() {
25:         return battery;
26:     }
27:
28: }
```

```
1: /*
2:  * Server.java
3:  * Dan Armendariz
4:  * Computer Science E-76
5:  *
6:  * Demonstrates class inheritance; Server is a subclass of Computer.
7:  *
8:  */
9:
10: class Server extends Computer {
11:
12:     private int rackHeight;
13:
14:     public Server(String mk, String mdl, int yr, double spd) {
15:         super(mk, mdl, yr, spd);
16:     }
17:
18:     public void setRackHeight(int height) {
19:         rackHeight = height;
20:     }
21:
22:     public int getRackHeight() {
23:         return rackHeight;
24:     }
25:
26: }
```