CSCI 145 Homework 1

Due Monday, Mar 20 2023

Name: Ivan Leung

Chapter 1

Exercise

Ex 1.6

a) 12 kb = 12000 bytes

b) 5 mb = 5000000 bytes

c) 3 GB = 3000000000 bytes

d) 2 TB = 2000000000000 bytes

Ex 1.15

a) Factorial is a valid identifier since it contains letters only.

b) anExtremelyLongIdentifierIfYouAskMe is a valid identifier since it contains letters only.

c) 2ndLevel is not a valid identifier since the first character is a number.

d) level2 is a valid identifier since the first character is a letter.

e) MAX_SIZE is a valid identifier since underscore is allowed in Java.

f) highest$ is a valid identifier since dollar sign is allowed in Java.

g) hook&ladder is not a valid identifier since ampersand is not allowed in Java.

Ex 1.16

a) q is not meaningful.

b) totVal is ambiguous.

c) theNextValueInTheList is too long to read.

Ex 1.17

It means that the upper case and lower case of the same letter are consider different characters.

Ex 1.19

We say that something is ambiguous when it has more than one possible meaning. If a programming statement is ambiguous, it is hard predict what the result could be.

Ex 1.20

a) logical error

b) run-time error

c) compile-time error

d) run-time error

e) logical error

f) compile-time error

Programming Project

pp 1.2

a) The public type test must be defined in its own file.

b) No error. String literal can have lower or upper case.

c) String literal is not properly closed by a double-quote.

d) String literal is not properly closed by a double-quote.

e) No error shown, but the program cannot compile in Eclipse.

f) The method bogus(String) is undefined for the type PrintStream.

g) Syntax error, insert ";" to complete block statements.

h) Syntax error, insert "}" to complete class body.

Chapter 2

Exercise

Ex 2.1

4 is an integer literal.

4.0 is a floating-point literal.

'4' is a character literal.

"4" is a string literal.

Ex 2.4

The statement has to be on a single line

System.out.println("To be or not to be, that is the question.");

Ex 2.6

He thrusts his fists

againstthe post

and still insists

he sees the “ghost”

Ex 2.7

size = 15

Ex 2.9

length = 1

Ex 2.10

++total;

total++;

total += 1;

total = total + 1;

Ex 2.11

a) iResult = 5

b) fResult = 5.0

c) iResult = 3

d) fResult = 3.0

e) fResult = 3.4

f) fResult = 1.3302034428794993

g) iResult = 0

h) fResult = 0.625

i) fResult = 0.625

j) fResult = 0.0

k) iResult = 3

l) fResult = 3.0

m) fResult = 0.0

n) iResult = 2

o) iResult = 6

p) iResult = 17

q) iResult = 0

Programming Project

pp 2.8

Source code below:

```
package csci_145.hw.hw1;

/*  Java Class: CSCI 145
Author: Ivan Leung
Class: Mon/Wed
Date: Mar 1 2023
Description:

I certify that the code below is my own work.

Exception(s): N/A

*/

import java.util.Scanner;

public class Seconds {
    public static void main(String[] args) {
        int hours, minutes, seconds, totalSeconds;
        Scanner scan = new Scanner(System.in);

        System.out.print("Enter the number of hours: ");
        hours = scan.nextInt();
        System.out.print("Enter the number of minutes: ");
        minutes = scan.nextInt();
        System.out.print("Enter the number of seconds: ");
        seconds = scan.nextInt();
        scan.close();
        System.out.println();

        totalSeconds = seconds + (minutes * 60) + (hours * 3600);

        System.out.print(hours + " hour(s) " + minutes + " minute(s) " + seconds
+ " seconds ");
```

```
        System.out.println("is equivalent to " + totalSeconds + " seconds.");
    }
}
```

Input/output below:

$ java Seconds.java

Enter the number of hours: 1

Enter the number of minutes: 28

Enter the number of seconds: 42

1 hour(s) 28 minute(s) 42 seconds is equivalent to 5322 seconds.

pp 2.9

Source cod below:

```
package csci_145.hw.hw1;

/*  Java Class: CSCI 145
Author: Ivan Leung
Class: Mon/Wed
Date: Mar 1 2023
Description:

I certify that the code below is my own work.

Exception(s): N/A

*/

import java.util.Scanner;

public class Secondsv2 {
    public static void main(String[] args) {
        int hours = 0, minutes = 0, seconds = 0, totalSeconds;
        Scanner scan = new Scanner(System.in);

        System.out.print("Enter the total number of seconds: ");
        totalSeconds = scan.nextInt();
        seconds = totalSeconds;
```

```
        while (seconds >= 3600) {
            seconds -= 3600;
            ++hours;
        }
        while (seconds >= 60) {
            seconds -= 60;
            ++minutes;
        }
        scan.close();
        System.out.println();

        System.out.println(totalSeconds + " seconds is equivalent to ");
        System.out.print( hours + " hour(s) " + minutes + " minute(s) " + seconds
+ " seconds.");
    }
}
```

Input/output below:

$ java Secondsv2.java

Enter the total number of seconds: 9999

9999 seconds is equivalent to

2 hour(s) 46 minute(s) 39 seconds.

Chapter 3

Exercise

Ex 3.3

String str = name.toUpperCase();

Ex 3.5

Quest for the Holy Grail quest for the holy zrail

Ex 3.7

a) 0 – 19

b) 1 – 8

c) 2 – 13

d) 10 – 44

e) -50 – 49

Ex 3.8

Random rand = new Random();

a) rand.nextInt(11);

b) rand.nexInt(401);

c) rand.nextInt(10) + 1;

d) rand.nextInt(400) + 1;

e) rand.nextInt(26) + 25;

f) rand.nextInt(26) – 10;

Ex 3.12

Code statements:

```
double val;
Scanner scan = new Scanner(System.in);
DecimalFormat decimal = new DecimalFormat("0.###");
System.out.print("Enter a double floating-point value: ");
val = scan.nextDouble();
scan.close();
System.out.println(decimal.format(Math.pow(val, 4)));
```

Input/output below:

```
Enter a double floating-point value: 3.5
150.062
```

Ex 3.15

a) Line line = new Line(60, 100, 30, 90);

b) Rectangle rectangle = new Rectangle(10, 10, 20, 100);

c) Circle circle = new Circle(50, 75, 30);

d) Ellipse ellipse = new Ellipse(150, 180, 50, 40);

Programming Project

pp 3.1

Source code below:

```
package hw1;

import java.util.Scanner;
import java.util.Random;

public class UserNames {

	public static void main(String[] args) {
		String firstName;
		String lastName;
		Scanner scan = new Scanner(System.in);
		Random rand = new Random();
		System.out.print("Enter your first name: ");
		firstName = scan.next();
		System.out.print("Enter your last name: ");
		lastName = scan.next();
		scan.close();
		System.out.println();
		System.out.println(firstName.charAt(0) + lastName.substring(0, 5) +
(rand.nextInt(90) + 10));
	}

}
```

Input/output below:

```
Enter your first name: Ivan
Enter your last name: Leung

ILeung37
```

pp 3.4

Source code below:

```
package hw1;

import java.util.Scanner;

public class WholeNumber {

	public static void main(String[] args) {
		double val;
		Scanner scan = new Scanner(System.in);
		System.out.print("Enter a floating-point value: ");
		val = scan.nextDouble();
		scan.close();
```

```
            System.out.println();
            System.out.println((int) Math.floor(val));
            System.out.println((int) Math.ceil(val));
        }

}
```

Input/output below:

```
Enter a floating-point value: 325.9724023

325
326
```

Chapter 4

Exercise

Ex 4.1

a) Superhero is the class and Superman is an object of the class.

b) Person is the class and Justin is an object of the class.

c) Pet is the class and Rover is an object of the class.

d) Magazine is the class and Time is an object of the class.

e) Holiday is the class and Christmas is an object of the class.

Ex 4.4

Some of attributes for a class called Course might have course name, course number, maximum number of students it can take, current number of students it has taken, course start time, and course end time.

Some of operations for a class called Course might have add student, remove student.

Ex 4.6

```
public int cube(int num) {
        int result = num * num * num;
        return result;
}
```

Ex 4.7

```
public int random100() {
        Random rand = new Random();
        int num = rand.nextInt(100) + 1;
```

```
        return num;
}
```

Ex 4.9

```
public Color randomColor() {
        Random rand = new Random();
        Color randColor = Color.rgb(rand.nextInt(256), rand.nextInt(256), rand.nextInt(256));
        return randColor;
}
```

Ex 4.11

```
public int getAge() {
        return age;
}
public void setAge(int age) {
        this.age = age;
}
```

Ex 4.12

Programming Project

pp 4.1

Source code below:

```
package hw1;

public class Counter {

	public Counter() {
		count = 0;
	}

	public void click() {
		++count;
	}

	public int getCount() {
		return count;
```

```
	}

	public void reset() {
		count = 0;
	}

	private int count;

}

package hw1;

public class CounterTest {

	public static void main(String[] args) {
		Counter test1 = new Counter();
		Counter test2 = new Counter();

		System.out.println("Initial count for Test 1: " +
test1.getCount());
		System.out.println("Initial count for Test 2: " +
test2.getCount());
		test1.click();
		for (int i = 0; i < 5; ++i) {
			test2.click();
		}
		System.out.println("Current count after click for Test 1: "
+ test1.getCount());
		System.out.println("Current count after 5 clicks for Test 2:
" + test2.getCount());
		test1.reset();
		test2.reset();
		System.out.println("Current count after reset for Test 1: "
+ test1.getCount());
		System.out.println("Current count after reset for Test 2: "
+ test2.getCount());
	}

}
```

Input/output below:

```
Initial count for Test 1: 0
Initial count for Test 2: 0
Current count after click for Test 1: 1
```

```
Current count after 5 clicks for Test 2: 5
Current count after reset for Test 1: 0
Current count after reset for Test 2: 0
```

pp 4.8

Source code below:

```
package hw1;

public class Flight {

	public Flight(String airlineName, String flightNumber, String
originCity, String destinationCity) {
		this.airlineName = airlineName;
		this.flightNumber = flightNumber;
		this.originCity = originCity;
		this.destinationCity = destinationCity;
	}

	public void setAirlineName(String airlineName) {
		this.airlineName = airlineName;
	}
	public void setFlightNumber(String flightNumber) {
		this.flightNumber = flightNumber;
	}
	public void setOriginCity(String originCity) {
		this.originCity = originCity;
	}
	public void setDestinationCity(String destinationCity) {
		this.destinationCity = destinationCity;
	}
	public String getAirlineName() {
		return this.airlineName;
	}
	public String getFlightNumber() {
		return this.flightNumber;
	}
	public String getOriginCity() {
		return this.originCity;
	}
	public String getDestinaitonCity() {
		return this.destinationCity;
	}
	public String toString() {
```

```
            return ("Airline Name: " + this.airlineName + ", Flight
Number: " + this.flightNumber + ", Origin City: " + this.originCity +
", Destination City: " + this.destinationCity);
      }

      private String airlineName;
      private String flightNumber;
      private String originCity;
      private String destinationCity;
}


package hw1;

public class FlightTest {

      public static void main(String[] args) {

            String name1 = "American Airline";
            String name2 = "Alaska Airline";
            String name3 = "Hawaiian Airline";
            String name4 = "Japan Airline";
            String city1 = "Los Angeles";
            String city2 = "New York";
            String city3 = "Alaska";
            String city4 = "Atlanta";
            String city5 = "Hawaiian";

            Flight flight1 = new Flight(name1, "AA1234", city1, city2);
            Flight flight2 = new Flight(name2, "AL1234", city3, city1);
            Flight flight3 = new Flight(name3, "HA1234", city3, city5);

            System.out.println("Initial flight data for Flight 1:\n" +
flight1);
            System.out.println("\nInitial flight data for Flight 2:\n" +
flight2);
            System.out.println("\nInitial flight data for Flight 3:\n" +
flight3);
            System.out.println();

            flight1.setAirlineName(name4);
            System.out.println("Flight 1's Airline changed to " +
flight1.getAirlineName());
            flight2.setFlightNumber("AL5678");
            System.out.println("Flight 2's Flight Number changed to " +
flight2.getFlightNumber());
```

```
            flight2.setDestinationCity(city2);
            System.out.println("Flight 2's destination city changed to "
+ flight2.getDestinaitonCity());
            flight3.setOriginCity(city4);
            System.out.println("Flight 3's origin city changed to " +
flight3.getOriginCity());
            System.out.println();

            System.out.println("Updated flight data for Flight 1:\n" +
flight1);
            System.out.println("\nUpdated flight data for Flight 2:\n" +
flight2);
            System.out.println("\nUpdated flight data for Flight 3:\n" +
flight3);
      }

}
```

Input/output below:

```
Initial flight data for Flight 1:
Airline Name: American Airline, Flight Number: AA1234, Origin City:
Los Angeles, Destination City: New York

Initial flight data for Flight 2:
Airline Name: Alaska Airline, Flight Number: AL1234, Origin City:
Alaska, Destination City: Los Angeles

Initial flight data for Flight 3:
Airline Name: Hawaiian Airline, Flight Number: HA1234, Origin City:
Alaska, Destination City: Hawaiian

Flight 1's Airline changed to Japan Airline
Flight 2's Flight Number changed to AL5678
Flight 2's destination city changed to New York
Flight 3's origin city changed to Atlanta

Updated flight data for Flight 1:
Airline Name: Japan Airline, Flight Number: AA1234, Origin City: Los
Angeles, Destination City: New York

Updated flight data for Flight 2:
Airline Name: Alaska Airline, Flight Number: AL5678, Origin City:
Alaska, Destination City: New York
```

Updated flight data for Flight 3:
Airline Name: Hawaiian Airline, Flight Number: HA1234, Origin City: Atlanta, Destination City: Hawaiian