Big Java Late Objects 2e

Chapter 1

R1.1

A well-designed computer program is easy to use without any special knowledge. For example, most people can learn to navigate webpages with only a few minutes of practice. On the other hand, programming a computer requires special knowledge about what the computer can fundamentally do and how you would communicate with it through a programming language.

R1.2

Typically program code is stored on a hard disk, CD/DVD disc, or in some other central location across a network. User data is often more likely to be stored on a local hard disk, although it can also be stored on a network or even a CD/DVD for backup storage.

R1.3

The monitor, speakers, and printer serve as the primary devices to give information to the user. The keyboard and mouse are the primary devices that take user input.

R1.4

It's very likely your cell phone is a programmable computer. If you can take pictures, send email/text messages, and/or surf the web with your phone, it is a programmable computer. If your cell phone can only send and receive phone calls, it is probably a single-function device.

R1.5

One advantage of Java over machine code is that Java statements are independent of the machine (computer) they are being executed on; machine code statements differ from one type of machine to the next. Another advantage of Java is that it is much more readable and understandable (by humans) than machine code.

R1.6

a)Solutions here will vary based on user and IDE preference. On a UNIX-based system using the Eclipse IDE you may see a path like

```
/home/nancy/JFE/src
```

While on a Microsoft Windows machine you might find a directory like:

```
C:\Users\nancy\Documents\JFE\src
```

b)Again, solutions can vary. On Unix using Eclipse you might see:

```
/home/nancy/JFE/bin
```

A Microsoft Windows machine might be:

```
C:\Users\nancy\Documents\JFE\bin
```

c)The answer to this question is dependent on the type of operating system and version of Java. On a Unix based system using Java 1.6 you might find `rt.jar` here:

```
/usr/lib/jvm/java-6-sun-1.6.0.13/jre/lib/rt.jar
```

While on a Microsoft Windows platform you might find it here:

```
C:\Program Files\Java\jdk1.6.0_10\jre
```

R1.7

The program prints the following:

```
39 + 3
42
```

Java interprets the statement `"39 + 3"` as a string and thus prints out the literal characters `39 + 3`. Java interprets the second statement `39 + 3` as an operation between two numbers, so it first calculates the value `39 + 3 = 42` then prints out the result `42`.

R1.8

```
HelloWorld
```

Because there are no spaces after the `System.out.print("Hello");` the next line prints `World` directly after `Hello` is printed.

R1.9

Java interprets the comma in the `println` method to mean that two strings are passed to `println`. It's likely the programmer meant to do this:

```
System.out.print("Hello, World!");
```

R1.10

Compile-time errors:

```
public class HelloPrinter
{
   public static void main(String[] args)
   {
      System.outprint("Hello, World");
/*             ^^ missing '.' */
   }
}
public class HelloPrinter
{
   public static void main(String[] args)
   {
      System.out.print("Hello, World);
/*                            ^^ missing '"' */
   }
}
public class HelloPrinter
{
   public static void main(String[] args)
   {
      System.out.print("Hello, World")
/*                               ^^ missing ";" */
   }
}
```

Run-time error:

```
public class HelloPrinter
{
   public static void main(String[] args)
   {
      System.out.print("Hello, Wrold");
   }
```

R1.11

Syntax errors are discovered by compiling your Java program; the Java compiler will report them directly. Logic errors are discovered by testing your program by running it and verifying the correct output; if the output is incorrect, you have a logic error.

R1.12

```
Start with the customer knowing how often they visit the cafeteria and the average price they pay for a meal.

Ask the cafeteria for cost of the discount card, the number of meals to be eaten to achieve the free meal,
and the period of time in which all meals must be consumed.

If cost of the card is greater than the average cost of a customer meal
  Deal is set to "no good"
Else
  If qty of meals to be eaten is greater than qty of customers meals in the time period then
    Deal is set to "no good"
  Else
    Deal is set to "good"
```

R1.13

```
Start with a year of value 0, a month of value 0, and a balance of $10,000
Repeat the following while the balance is greater than 0
  Multiply the balance by 1.005.
  Subtract 500 from the balance.
  Add one to month.
  If the month is 12
    Set month to 0.
    Add one to year.
Year has the answer.
```

R1.14

If the starting balance is large enough and the interest rate large enough such that the first month's calculation of interest exceeds the monthly expenses, then you will never deplete your account and the algorithm will never terminate.

```
Modified algorithm:
Ask the user for balance, interest rate, and monthly expenses.
If balance x interest is greater than monthly expenses
  Tell the user you're in good financial shape and quit.
Otherwise, start with a year of value 0, a month of value 0 and a balance of $10,000
Repeat the following while the balance is greater than 0
  Multiply the balance by (1 + (interest rate times .01 divided by 12)).
  Subtract monthly expenses from the balance.
  Add one to month
  If the month is 12
    Set month to 0.
    Add one to year.
Year has the answer.
```

R1.15

```
Calculate the overall surface area of the house without windows and doors.

surfaceArea = (2 x width x height) + (2 x length x height)
 - (number of windows x windowWidth x windowHeight)
 - (number of doors x doorWidth x doorHeight)
```

R1.16

The computer cannot accurately predict what the price of gas will be on a daily basis (or know what day you will go to the gas station) and how many actual miles you will drive each year.

R1.17

Every day at 5 P.M. please do the following:

1)Insert the USB memory stick into the computer.

2)Create a new directory on the USB memory stick entitled "BACKUP-DATE" where "DATE" is replaced with the current date.

3)Copy the files from the "My Documents" folder on the computer to the folder you just created on the USB stick.

4)Double check to make sure the new directory contains the files you copied. If the folder is empty, something is wrong. It's possible you backed up to the wrong directory. Try it again and be careful which directory you copy into.

5)Once you verified the copy is complete, remove the USB stick and store it someplace safe.

6)Thank you!

R1.18

```
Get the orange juice from the refrigerator
Get the eggs from the refrigerator
Get the bacon from the refrigerator
Make the pancake batter
For each person eating breakfast:
   Crack an egg on the griddle
   Pour two pancakes on the griddle
   Place two slices of bacon on the griddle
   Pour a glass of orange juice
   Dish up a plate of cooked food
Call the family to breakfast
```

R1.19

```
close enough = 0.001
a = value from user to be square rooted
first guess = a / 2
second guess = (first guess + (a / first guess)) / 2
repeat
   first guess = second guess
   second guess = (first guess + (a / first guess)) / 2
until (first guess - second guess) <= close enough

second guess holds the square root of a
```

Chapter 2

R2.1

This is somewhat ambiguous. Are the variables local variables, or class variables? I assume class variables:

```
public static final int DAYS_IN_WEEK = 7;
public int daysToSemesterEnd;
public static final double CENTIMETERS_IN_INCH = 2.54;
public double tallestHeight;
```

R2.2

The value of `mystery` is equal to 0 after the statements are executed. In the first statement (line 1), `mystery` is initialized to a value of 1. In the assignment statement on line 2, `mystery` is set to -1. Finally, `mystery` is set to 0 in line 3.

R2.3

The variable `mystery` is being declared twice, first in line 1 and then again in line 2. A variable can only be initialized once. (If you remove the reserved word `int` on line 3, the statements will work just fine, and will set `mystery` to -3.)

R2.4

a) `double s = s0 +(v0 * t) + (.5 * g * Math.pow(t,2));`