

Software Engineering

(Chap. 1)

Object-Centered Design

When we prepare a program, the experience can be just like composing poetry or music ... My claim is that it is possible to write grand programs, noble programs, truly magnificent ones! ... Computer programming is an art.

- Donald Knuth, "Programming as an Art", 1974

Problem Solving

Let's solve this *temperature-conversion problem*:

Write a program that, given a temperature in Celsius, displays that temperature in Fahrenheit.

5 Phases of Software Life Cycle:

- Problem Analysis and Specification
- Design ← **OCD (Object-Centered Design)**
- Implementation (Coding)
- Testing, Execution and Debugging
- Maintenance

2

OCD in a Nutshell

- Behavior
- Objects
- Operations
- Algorithm

Using
OCD

Behavior

A. Describe the desired behavior of the program:

Our program should display a prompt for the Celsius temperature on the screen, read that temperature from the keyboard, compute the corresponding Fahrenheit temperature, and display the result, along with a descriptive label on the screen.

4

Using
OCD

Objects

B. Identify the *nouns* in the behavioral description:

Our program should display a prompt for the Celsius temperature on the screen, read that temperature from the keyboard, compute the corresponding Fahrenheit temperature, and display that temperature, along with a descriptive label on the screen.

These make up the *objects* in our program.

5

Using
OCD

Operations

C. Identify the *verbs* in the behavioral description:

Our program should display a prompt for the Celsius temperature on the screen, read that temperature from the keyboard, compute the corresponding Fahrenheit temperature, and display that temperature, along with a descriptive label on the screen.

These make up the *operations* in our program.

6

Using
OCD

Algorithm

D. Organize the objects and operations into a sequence of steps that solves the problem, called an *algorithm*.

1. Ask *the screen* to display a *prompt* for the *Celsius temperature* on the screen.
2. Ask *the keyboard* to read *the temperature*.
3. Compute *the Fahrenheit temperature* from *the Celsius temperature*.
4. Ask *the screen* to display *the Fahrenheit temperature*, plus an *informative label* on the screen.

7



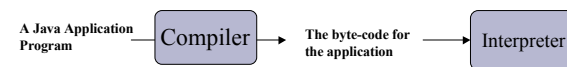
Don't even think about coding until
you're sure of your algorithm.

Coding

- We now translate the algorithm into an HLL.
- Use the features of the Java language to:
 - represent our objects
 - perform our operations
- Java provides two implementation methods:
 - Java _____
 - Java _____ (Sec. 1.3)

Java Applications

- Programming languages traditionally allow programmers to write stand-alone applications.
- In Java, these are called *applications*.



Representing Objects

Determine a _____ and _____ for each object:

Object	Java Type	Name
the program	Object	<i>Temperature</i>
a prompt	String	--
the Celsius temp.	double	<i>celsius</i>
the screen	Screen	<i>theScreen</i>
the keyboard	Keyboard	<i>theKeyboard</i>
the Fahrenheit temp.	double	<i>fahrenheit</i>
a label	String	--

Performing Operations

Identify the Java operator to perform a given operation, if there is one...

Operation	Library?	Name
Display a string	ann.easyio	print()
Read a double	ann.easyio	readDouble()
Compute fahrenheit	--	--
Display a double	ann.easyio	print()

To compute the Fahrenheit temperature, we need to find the Celsius to Fahrenheit formula in a reference book...

$$fahrenheit = (9/5)celsius + 32$$

Operations (Revised)

We can now add more detail to the computation:

Operation	Library?	Name
Display a string	ann.easyio	print()
Read a double	ann.easyio	readDouble()
Compute fahrenheit	--	--
Divide two ints	built-in	/
Multiply two doubles	built-in	*
Add two doubles	built-in	+
Display a double	ann.easyio	print()

The Code

Objects?
Operations?

```

/* Temperature.java converts Celsius
 * temperatures to Fahrenheit.
 * Author: Keith Vander Linden
 * Date: 2 June 2001
 * Modified: L. Nyhoff, Sept. 2002
 */

import ann.easyio.*; // Screen & Keyboard classes

class Temperature extends Object
{
    public static void main(String [] args)
    {
        Screen theScreen = new Screen();
        theScreen.print("Welcome to the temperature converter!\n" +
            "Please enter the temperature in Celsius: ");

        Keyboard theKeyboard = new Keyboard();
        double celsius = theKeyboard.readDouble();

        double fahrenheit = ((9.0/5.0)*celsius) + 32;

        theScreen.print(celsius + " degrees Celsius is " +
            fahrenheit + " degrees Fahrenheit.\n" +
            "It's been a pleasure!\n");
    }
}

```

COMPILATION HINT

- Use the tools available to you.
- Specifically, use `M-x compile` in xemacs.
 - Use `C-x `` to run through the error messages.
 - Xemacs will automatically go to the next message.
 - It will even load in the proper file and go to the offending line.
 - Why do you want to do this work?
 - Run only one session of xemacs; open multiple files into the same session.
 - If you aren't comfortable or used to it, practice.

Documentation

Always begin a file with an *opening comment* similar to this..

```

/* Temperature.java converts Celsius
 * temperatures to Fahrenheit.
 * Author: Keith Vander Linden
 * Date: 2 June 2001
 * Modified: L. Nyhoff, Sept. 2002
 */

import ann.easyio.*; // Screen & Keyboard classes

class Temperature extends Object
{
    public static void main(String [] args)
    {
        Screen theScreen = new Screen();
        theScreen.print("Welcome to the temperature converter!\n" +
            "Please enter the temperature in Celsius: ");

        Keyboard theKeyboard = new Keyboard();
        double celsius = theKeyboard.readDouble();

        double fahrenheit = ((9.0/5.0)*celsius) + 32;

        theScreen.print(celsius + " degrees Celsius is " +
            fahrenheit + " degrees Fahrenheit.\n" +
            "It's been a pleasure!\n");
    }
}

```

3 Kinds of Comments:

- /*...*/ multi-line comment
- // single-line comment
- indicate what classes of a package are used.
- explain obscure code
- /**...*/ Javadoc multi-line comment
- javadoc program extracts these into special documentation format
- see Java's API (Section 2.4)
- Compiler ignores all comments

```

/* Temperature.java converts Celsius
 * temperatures to Fahrenheit.
 * Author: Keith Vander Linden
 * Date: 2 June 2001
 * Modified: L. Nyhoff, Sept. 2002
 */

import ann.easyio.*; // Screen & Keyboard classes

class Temperature extends Object
{
    public static void main(String[] args)
    {
        Screen theScreen = new Screen();
        theScreen.print("Welcome to the temperature converter!\n" +
            "Please enter the temperature in Celsius: ");

        Keyboard theKeyboard = new Keyboard();
        double celsius = theKeyboard.readDouble();

        double fahrenheit = ((9.0/5.0)*celsius) + 32;

        theScreen.print(celsius + " degrees Celsius is " +
            fahrenheit + " degrees Fahrenheit.\n" +
            "It's been a pleasure!\n");
    }
}

```

Import Section
This loads the Java packages that we need.

Packages:

- Groups of related classes e.g., easyio
- Often grouped into *libraries* e.g., ann
- Java has > 1600 classes grouped into several packages -- see its API

Use

`import package_name.*;`
to make classes in a package easily accessible

```

/* Temperature.java converts Celsius
 * temperatures to Fahrenheit.
 * Author: Keith Vander Linden
 * Date: 2 June 2001
 * Modified: L. Nyhoff, Sept. 2002
 */

import ann.easyio.*; // Screen & Keyboard classes

class Temperature extends Object
{
    Screen theScreen = new Screen();
    theScreen.print("Welcome to the temperature converter!\n" +
        "Please enter the temperature in Celsius: ");

    Keyboard theKeyboard = new Keyboard();
    double celsius = theKeyboard.readDouble();

    double fahrenheit = ((9.0/5.0)*celsius) + 32;

    theScreen.print(celsius + " degrees Celsius is " +
        fahrenheit + " degrees Fahrenheit.\n" +
        "It's been a pleasure!\n");
}

```

The Class Definition
Java programs are classes built upon existing classes

Form:

```

class Class_name extends Object
{
    public static void main(String[] args)
    {
        statements
    }
}

```

- Use meaningful class name
- Capitalize it
- Save program as `Class_name.java`

```

/* Temperature.java converts Celsius
 * temperatures to Fahrenheit.
 * Author: Keith Vander Linden
 * Date: 2 June 2001
 * Modified: L. Nyhoff, Sept. 2002
 */

import ann.easyio.*; // Screen & Keyboard classes

class Temperature extends Object
{
    public static void main(String [] args)
    {
        Screen theScreen = new Screen();
        theScreen.print("Welcome to the temperature converter!\n" +
            "Please enter the temperature in Celsius: ");

        Keyboard theKeyboard = new Keyboard();
        double celsius = theKeyboard.readDouble();

        double fahrenheit = ((9.0/5.0)*celsius) + 32;

        theScreen.print(celsius + " degrees Celsius is " +
            fahrenheit + " degrees Fahrenheit.\n" +
            "It's been a pleasure!\n");
    }
}

```

The Main Method
Java applications begin execution by running `main()`

```

/* Temperature.java converts Celsius
 * temperatures to Fahrenheit.
 * Author: Keith Vander Linden
 * Date: 2 June 2001
 * Modified: L. Nyhoff, Sept. 2002
 */

import ann.easyio.*; // Screen & Keyboard classes

class Temperature extends Object
{
    public static void main(String [] args)
    {
        Screen theScreen = new Screen();
        theScreen.print("Welcome to the temperature converter!\n" +
            "Please enter the temperature in Celsius: ");

        Keyboard theKeyboard = new Keyboard();
        double celsius = theKeyboard.readDouble();

        double fahrenheit = ((9.0/5.0)*celsius) + 32;

        theScreen.print(celsius + " degrees Celsius is " +
            fahrenheit + " degrees Fahrenheit.\n" +
            "It's been a pleasure!\n");
    }
}

```

Step 1
Print a friendly message.

```

/* Temperature.java converts Celsius
 * temperatures to Fahrenheit.
 * Author: Keith Vander Linden
 * Date: 2 June 2001
 * Modified: L. Nyhoff, Sept. 2002
 */

import ann.easyio.*; // Screen & Keyboard classes

class Temperature extends Object
{
    public static void main(String [] args)
    {
        Screen theScreen = new Screen();
        theScreen.print("Welcome to the temperature converter!\n" +
            "Please enter the temperature in Celsius: ");

        Keyboard theKeyboard = new Keyboard();
        double celsius = theKeyboard.readDouble();

        double fahrenheit = ((9.0/5.0)*celsius) + 32;

        theScreen.print(celsius + " degrees Celsius is " +
            fahrenheit + " degrees Fahrenheit.\n" +
            "It's been a pleasure!\n");
    }
}

```

Step 2
Read the Celsius temperature.

```

/* Temperature.java converts Celsius
 * temperatures to Fahrenheit.
 * Author: Keith Vander Linden
 * Date: 2 June 2001
 * Modified: L. Nyhoff, Sept. 2002
 */

import ann.easyio.*; // Screen & Keyboard classes

class Temperature extends Object
{
    public static void main(String [] args)
    {
        Screen theScreen = new Screen();
        theScreen.print("Welcome to the temperature converter!\n" +
            "Please enter the temperature in Celsius: ");

        Keyboard theKeyboard = new Keyboard();
        double celsius = theKeyboard.readDouble();

        double fahrenheit = ((9.0/5.0)*celsius) + 32;

        theScreen.print(celsius + " degrees Celsius is " +
            fahrenheit + " degrees Fahrenheit.\n" +
            "It's been a pleasure!\n");
    }
}

```

Step 3
Calculate the Fahrenheit temperature.

```

/* Temperature.java converts Celsius
 * temperatures to Fahrenheit.
 * Author: Keith Vander Linden
 * Date: 2 June 2001
 * Modified: L. Nyhoff, Sept. 2002
 */

import ann.easyio.*; // Screen & Keyboard classes

class Temperature extends Object
{
    public static void main(String [] args)
    {
        Screen theScreen = new Screen();
        theScreen.print("Welcome to the temperature converter!\n" +
            "Please enter the temperature in Celsius: ");

        Keyboard theKeyboard = new Keyboard();
        double celsius = theKeyboard.readDouble();

        double fahrenheit = ((9.0/5.0)*celsius) + 32;

        theScreen.print(celsius + " degrees Celsius is " +
            fahrenheit + " degrees Fahrenheit.\n" +
            "It's been a pleasure!\n");
    }
}

```

Step 4
Display the results.

Running the Program

You can now run your application:

Welcome to the temperature converter!
Please enter the temperature in Celsius: 20
20.0 degrees Celsius is 68.0 degrees Fahrenheit.
It's been a pleasure!

Testing

- Getting your code to compile and run is not the main goal.
- Rather, your goal is to produce a useful program that is:
 - correct
 - efficient
 - readable
 - usable

Running Test Cases

Do many tests on “interesting” data points.

```
Welcome to the temperature converter!  
Please enter the temperature in Celsius: 0  
0.0 degrees Celsius is 32.0 degrees Fahrenheit.  
It's been a pleasure!
```

```
Welcome to the temperature converter!  
Please enter the temperature in Celsius: -17.78  
-17.78 degrees Celsius is -0.00400000000000048885  
degrees Fahrenheit.  
It's been a pleasure!
```

Writing Code

- Comments (`/* ... */` or `//`) are completely ignored by the compiler.
- Whitespace matters only two places:
 - Inside quotes, and
 - In comment delimiters (`/*`, `*/`, `//`).
- A semi-colon indicates the end of a program statement.

COMPILATION HINT

- Forgotten or extra semi-colons confuse the compiler.
- The compiler never recognizes that there's a semi-colon problem; it usually won't say what's wrong.
- Usually a "parse error" is an indication that the *previous* line of code is missing a semi-colon or has one it shouldn't.