

> Summary and Outcomes

In this unit, students will learn to convert an integer value for month to a String value. To do this, they will implement an algorithm as part of a class called `MonthName`. They will then develop an applet called `MonthNameApplet`, into which users can input a given month as an integer value (for example, 1 = January). The program will return the alphabetic, String, version of the month (“January,” “February,” and so on). Finally, students will convert the applet into an HTML page. Once the program is up and running, students can modify and improve upon the applet, adding color or other components. They may also add color and other improvements to the web (HTML) page.

> Student Assessment

How well do students perform each of the following tasks:

- Code and compile the `MonthName` class
- Code and compile the `MonthNameApplet` class
- Generate the web page with an embedded `MonthNameApplet`

> Prerequisite Knowledge and Skills

BlueJ:

- Creating projects
- Adding new classes
- Running applets
- Converting applets into HTML web pages

Java:

- Understanding the methods required for applets
- Using switch statements

HTML:

- Understanding how to create a basic web page

> This Activity Targets the Following:

Pre-Java	Java Programming
<input type="checkbox"/> Hardware Basics	<input checked="" type="checkbox"/> Applet Programming
<input checked="" type="checkbox"/> Software Basics	<input type="checkbox"/> Subroutine Programming
<input type="checkbox"/> Networks and Servers	<input type="checkbox"/> Full Scale Programming
<input checked="" type="checkbox"/> HTML	
<input type="checkbox"/> Action Scripting	
<input type="checkbox"/> Java Scripting	

> Curriculum-Framing Questions

Essential Question

- What is the connection between applets and HTML?
- How does a web page incorporate an applet?

Unit Questions

- What method is used to convert the representation of a month from an integer to a String?
- What components are needed in this applet?

Sample Content Questions

- How does the applet interact with the LeapYear class?
- What would make this applet more interesting?
- What other methods could be used to do this conversion?

> Targeted Content Standards, Benchmarks, or State Frameworks

California State Standards:

Mathematics:

- **Algebra I: Standard 25.0** – Students use properties of the number system to judge the

International Society of Technology Education (ISTE): Information Literacy Standards:

- **Standard 1** – The student who is information literate accesses

Sun, Sun Microsystems, the Sun Logo are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and other countries. This Java activity was developed for the The Sun Foundation by the East Union High School 2004.

validity of results, to justify each step of a procedure, and to prove or disprove statements.

- **Algebra I: Standard 25.1** – Students use properties of numbers to construct simple, valid arguments (direct and indirect) for, or formulate counterexamples to, claimed assertions.
- **Algebra I: Standard 25.2** – Students judge the validity of an argument according to whether the properties of the real number system and the order of operations have been applied correctly at each step.

information efficiently and effectively.

- **Standard 2** – The student who is information literate evaluates information critically and competently.

Technology Foundation Standards for Students:

- **Standard 3** – Students use technology tools to enhance learning, increase productivity, and promote creativity. Students use productivity tools to collaborate in constructing technology-enhanced models, prepare publications, and produce other creative works.

Mathematics Standards:

- **Standard 1: Number and Operation** – Mathematics instructional programs should foster the development of number and operation sense so that all students
 - Understand numbers, ways of representing numbers, relationships among numbers, and number

- systems
- Understand the meaning of operations and how they relate to each other
- Use computational tools and strategies fluently and estimate appropriately
- **Standard 2: Patterns, Functions, and Algebra – Mathematics** instructional programs should include attention to patterns, functions, symbols, and models so that all students
 - Understand various types of patterns and functional relationships
 - Use symbolic forms to represent and analyze mathematical situations and structures
 - Use mathematical models and analyze change in both real and abstract contexts

> Materials and Resources Required for Activity

Equipment	<ul style="list-style-type: none">• Computer for each student with BlueJ and Java SDK installed
Consumable Supplies	<ul style="list-style-type: none">• Diskettes• Printed output for prototype
Textbooks/Lesson Guides	<ul style="list-style-type: none">• Any of the better high school–level Java textbooks• Handouts of lessons
Technology	<ul style="list-style-type: none">• Windows-based machines running Windows 98 SE or higher
Internet Resources	<ul style="list-style-type: none">• http://java.sun.com/j2se/1.5.0/docs/api/• http://www.textpad.com/
Others	<ul style="list-style-type: none">• Text editor such as TextPad Or NotePad

> Activity/Lesson Plan Outline

Overview: In this unit, students will develop the algorithm to convert a numerical representation of a month to a String representation. They will then design and code an applet into which the user can enter the months to be converted. The applet will use the `MonthName` class to perform the conversion and will output the result. Once they have completed both of these tasks, students will create HTML code to generate a web page with this applet embedded.

Begin by creating a project called `MonthName` and within that project create a class called `MonthName`. What follows is a simple version of the `MonthName` code. Code and compile the following:

```
/**
 * Write a description of class MonthName here.
 *
 * @author (your name)
 * @version (a version number or a date)
 */
public class MonthName
{
    // instance variables - replace the example below with your
    own
    private int intMonth;
    private String strMonth;

    /**
     * Constructor for objects of class MonthName
     */
    public MonthName(int mon)
    {
        // initialise instance variables
        intMonth = mon;
    }
    public String name()
    {
        switch (intMonth)
        {
            case 1:
                strMonth = "January";
                break;
            case 2:
                strMonth = "February";
```

Sun, Sun Microsystems, the Sun Logo are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and other countries. This Java activity was developed for the The Sun Foundation by the East Union High School 2004.

```

        break;
    case 3:
        strMonth = "March";
        break;
    case 4:
        strMonth = "April";
        break;
    case 5:
        strMonth = "May";
        break;
    case 6:
        strMonth = "June";
        break;
    case 7:
        strMonth = "July";
        break;
    case 8:
        strMonth = "August";
        break;
    case 9:
        strMonth = "September";
        break;
    case 10:
        strMonth = "October";
        break;
    case 11:
        strMonth = "November";
        break;
    case 12:
        strMonth = "December";
        break;
    default:
        strMonth = "Invalid";
        break;
    }
    return strMonth;
}
}
} // end of MonthName class

```

The algorithm for converting from integer to `String` representation equates numbers and months as follows:

```

-> January
February

```

```

2 ->

```

Sun, Sun Microsystems, the Sun Logo are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and other countries. This Java activity was developed for the The Sun Foundation by the East Union High School 2004.

```

-> March
-> May
-> July
August
9 -> September
    11 -> November
December
    4 -> April
    6 -> June
    8 ->
    10 -> October
    12 ->

```

All other numeric values -> Invalid

Create an applet that asks the user for a numerical representation of a month and have it respond with the String corresponding to that number. Sample code for such an applet is illustrated as follows:

```

import java.awt.*;
import javax.swing.*;
import java.awt.event.*;

/**
 * Class MonthNameApplet - write a description of the class here
 *
 * @author (your name)
 * @version (a version number)
 */
public class MonthNameApplet extends JApplet implements
ActionListener
{
    // instance variables - replace the example below with your own
    private JLabel lblMonth;
    private JTextField txtMonth;
    private JLabel lblMonthName;
    private JLabel lblName;
    private JButton btnCalculate;
    private MonthName testMonth;

    /**
     * Called by the browser or applet viewer to inform this JApplet
that it
     * has been loaded into the system. It is always called before
the first
     * time that the start method is called.
     */
    public void init()
    {
        // this is a workaround for a security conflict with some
browsers
        // including some versions of Netscape & Internet Explorer
which do
        // not allow access to the AWT system event queue which
JApplets do

```

Sun, Sun Microsystems, the Sun Logo are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and other countries. This Java activity was developed for the The Sun Foundation by the East Union High School 2004.

```

        // on startup to check access. May not be necessary with
your browser.
        JRootPane rootPane = this.getRootPane();
        rootPane.putClientProperty("defeatSystemEventQueueCheck",
            Boolean.TRUE);
        lblMonth = new JLabel("Month");
        txtMonth = new JTextField("        ");
        lblMonthName = new JLabel("Name");
        lblName = new JLabel("        ");
        btnCalculate = new JButton("Calculate");

        btnCalculate.addActionListener(this);

        setLayout(new FlowLayout());
add(lblMonth);
        add(txtMonth);
        add(lblMonthName);
        add(lblName);

        add(btnCalculate);

        // provide any initialisation necessary for your JApplet
    }

    /**
    * Called by the browser or applet viewer to inform this JApplet
that it
    * should start its execution. It is called after the init
method and
    * each time the JApplet is revisited in a Web page.
    */
    public void start()
    {
        // provide any code required to run each time
        // web page is visited
    }

    /**
    * Called by the browser or applet viewer to inform this JApplet
that
    * it should stop its execution. It is called when the Web page
that
    * contains this JApplet has been replaced by another page, and
also
    * just before the JApplet is to be destroyed.
    */
    public void actionPerformed(ActionEvent evt)
    {
        // provide any code that needs to be run when page

```

```

        // is replaced by another page or before JApplet is
destroyed
        String strMonth;
        int month;
        strMonth = txtMonth.getText().trim();
        month = Integer.parseInt(strMonth);
        testMonth = new MonthName(month);

        String strMonthName = testMonth.name();
        lblName.setText(strMonthName);

    }
} //end of MonthNameApplet class

```

Next, code and debug this applet.

Finally, convert the applet into an HTML file. If you are using BlueJ, an HTML file will automatically be generated when the MonthNameApplet is run. If you are using a different compiler, use a text editor such as TextPad (NotePad will also work) to create the HTML file as follows:

```

<html>
  <head>
    <title>MonthNameApplet Applet</title>
  </head>
  <body>
    <h1>MonthNameApplet Applet</h1>
    <hr>
    <applet code="MonthNameApplet.class"
            width=500
            height=500
            codebase="."
            archive="file:/C:/Desktop%20clutter/chnutil.jar"
            alt="Your browser understands the <APPLET> tag but
isn't
                running the applet, for some reason."
    >
        Your browser is ignoring the <APPLET> tag!
    </applet>
    <hr>
  </body>
</html>

```

Note: When creating an HTML file that includes an applet, make sure that both the MonthNameApplet **class file** (MonthNameApplet.class) and the MonthName **class file** (MonthName.class) are in the same location as the HTML file, that is, in the

Sun, Sun Microsystems, the Sun Logo are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and other countries. This Java activity was developed for the The Sun Foundation by the East Union High School 2004.

same directory or folder.

> Pacing/Timeline

- Code and compile `MonthName` class: 1 day
- Code and compile `MonthNameApplet` class: 1 day
- Convert applet into web page and test: 1 day

Teacher Reflection (For example, what worked well in this unit? What would you change if you were to teach it again?)

- Were students able to code and compile the `MonthName` class?
- Were they able to code and compile the `MonthNameApplet` class?
- Were they able to convert the applet into an HTML document?
- Did they improve the applet code and experiment with color and placement of components?
- Were they able to run and improve upon the HTML code?

> Student Handout: Java Programming

Month Name

In this activity, you will implement an algorithm to convert a numeric (integer) representation for a month into an alphabetic (String) representation. Begin by creating a project called `MonthName` and within that project a class called `MonthName`. What follows is a simple version of the `MonthName` code. Code and compile the following:

```
/**
 * Write a description of class MonthName here.
 *
 * @author (your name)
 * @version (a version number or a date)
 */
public class MonthName
{
    // instance variables - replace the example below with your
own
    private int intMonth;
    private String strMonth;

    /**
     * Constructor for objects of class MonthName
     */
    public MonthName(int mon)
    {
        // initialise instance variables
        intMonth = mon;
    }

    public String name()
    {
        switch (intMonth)
        {
            case 1:
                strMonth = "January";
                break;
            case 2:
                strMonth = "February";
                break;
            case 3:
                strMonth = "March";
                break;
        }
    }
}
```

Sun, Sun Microsystems, the Sun Logo are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and other countries. This Java activity was developed for the The Sun Foundation by the East Union High School 2004.

```

        case 4:
            strMonth = "April";
            break;
        case 5:
            strMonth = "May";
            break;
        case 6:
            strMonth = "June";
            break;
        case 7:
            strMonth = "July";
            break;
        case 8:
            strMonth = "August";
            break;
        case 9:
            strMonth = "September";
            break;
        case 10:
            strMonth = "October";
            break;
        case 11:
            strMonth = "November";
            break;
        case 12:
            strMonth = "December";
            break;
        default:
            strMonth = "Invalid";
            break;
    }
    return strMonth;
}
}
}

} // end of MonthName class

```

Once this is coded and compiled, add to the `MonthName` project a new class called `MonthNameApplet`. Sample code for this applet is illustrated as follows:

```

import java.awt.*;
import javax.swing.*;
import java.awt.event.*;

/**
 * Class MonthNameApplet - write a description of the class here
 */

```

Sun, Sun Microsystems, the Sun Logo are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and other countries. This Java activity was developed for the The Sun Foundation by the East Union High School 2004.

```

* @author (your name)
* @version (a version number)
*/
public class MonthNameApplet extends JApplet implements
ActionListener
{
    // instance variables - replace the example below with your own
    private JLabel lblMonth;
    private JTextField txtMonth;
    private JLabel lblMonthName;
    private JLabel lblName;
    private JButton btnCalculate;
    private MonthName testMonth;

    /**
     * Called by the browser or applet viewer to inform this JApplet
that it
     * has been loaded into the system. It is always called before
the first
     * time that the start method is called.
     */
    public void init()
    {
        // this is a workaround for a security conflict with some
browsers
        // including some versions of Netscape & Internet Explorer
which do
        // not allow access to the AWT system event queue which
JApplets do
        // on startup to check access. May not be necessary with
your browser.
        JRootPane rootPane = this.getRootPane();
        rootPane.putClientProperty("defeatSystemEventQueueCheck",
Boolean.TRUE);
        lblMonth = new JLabel("Month");
        txtMonth = new JTextField("          ");
        lblMonthName = new JLabel("Name");
        lblName = new JLabel("          ");
        btnCalculate = new JButton("Calculate");

        btnCalculate.addActionListener(this);

        setLayout(new FlowLayout());
        add(lblMonth);
        add(txtMonth);
        add(lblMonthName);
        add(lblName);

        add(btnCalculate);
    }
}

```

```

        // provide any initialisation necessary for your JApplet
    }

    /**
     * Called by the browser or applet viewer to inform this JApplet
that it
     * should start its execution. It is called after the init
method and
     * each time the JApplet is revisited in a Web page.
     */
    public void start()
    {
        // provide any code required to run each time
        // web page is visited
    }

    /**
that
     * Called by the browser or applet viewer to inform this JApplet
that
     * it should stop its execution. It is called when the Web page
also
     * contains this JApplet has been replaced by another page, and
     * just before the JApplet is to be destroyed.
     */
    public void actionPerformed(ActionEvent evt)
    {
        // provide any code that needs to be run when page
        // is replaced by another page or before JApplet is
destroyed
        String strMonth;
        int month;
        strMonth = txtMonth.getText().trim();
        month = Integer.parseInt(strMonth);
        testMonth = new MonthName(month);

        String strMonthName = testMonth.name();
        lblName.setText(strMonthName);
    }
} // end of MonthNameApplet class

```

Next, compile and run the applet. Finally, convert the applet into an HTML file. If you are using BlueJ, an HTML file will automatically be generated when the MonthNameApplet is run. If you are using a different compiler, create a web (HTML)

Sun, Sun Microsystems, the Sun Logo are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and other countries. This Java activity was developed for the The Sun Foundation by the East Union High School 2004.

page as follows, making sure that both `MonthName.class` and `MonthNameApplet.class` files are located in the same directory (folder) as the HTML document.

```
<html>
  <head>
    <title>MonthNameApplet Applet</title>
  </head>
  <body>
    <h1>MonthNameApplet Applet</h1>
    <hr>
    <applet code="MonthNameApplet.class"
            width=500
            height=500
            codebase="."
            archive="file:/C:/Desktop%20clutter/chnutil.jar"
            alt="Your browser understands the <APPLET> tag but
            isn't running the applet, for some reason."
    >

        Your browser is ignoring the <APPLET> tag!
    </applet>
    <hr>
  </body>
</html>
```

Once you have created the HTML page, run it to verify that the applet works correctly.

> YOUR JOB:

- Code and compile the `MonthName` file
- Code and compile the `MonthNameApplet` file
- Convert the applet to HTML

> EXTRAS YOU CAN ADD:

- Add color and other features to the applet
- Modify and improve the web (HTML) page
- Use an array of names to implement the conversion

Sun, Sun Microsystems, the Sun Logo are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and other countries. This Java activity was developed for the The Sun Foundation by the East Union High School 2004.