

LESSON PLAN: Computer Programming

KEY CONCEPT: Debugging and Programming

The student will identify errors in a computer program and categorize the error, rewrite the erroneous code with modifications, rerun the code and evaluate for accuracy and success of the intended simulation.

PERFORMANCE OBJECTIVES:

Students will be able to:

- identify the 5 types of programming errors
- make modifications to a program so it is error-free

PATHWAY STANDARDS ADDRESSED: (List)

Information Technology Pathway Standard

7.4 Diagnose and solve application software problems

Three Foundation Standards

English Grades Nine and Ten

1.0 Word Analysis, Fluency, and Systematic Vocabulary Development

Math Grade 7

1.1 Analyze problems by identifying relationships, distinguishing relevant from irrelevant information, identifying missing information, sequencing and prioritizing information, and observing patterns.

ACADEMIC SKILLS REINFORCED: (List)

Language Arts Math Science Social Studies Other _____

FOUNDATION OR ESLRS ADDRESSED:

<input checked="" type="checkbox"/> Communication Skills	<input type="checkbox"/> Career Planning & Management
<input checked="" type="checkbox"/> Technology Skills	<input checked="" type="checkbox"/> Problem-solving Skills
<input type="checkbox"/> Health, Safety & Environment	<input checked="" type="checkbox"/> Accountability & Adaptability
<input type="checkbox"/> Ethics & Legal Responsibility	<input checked="" type="checkbox"/> Leadership & Teamwork

MATERIALS:

- Teacher provided program (EvenOdd.java) with a simple program error. EvenOdd.java (Solution) file has the corrected version of the program.
- Additional program (FahrenheitGUI.java) provided by teacher with some simple program errors. FahrenheitGUI.java (Solution) file has the corrected version of the program.
- NOTE: To view Java files, open them in Notepad or any Java IDE.
- Rubric for evaluating student performance

MOTIVATION/OPENER:

Students will play a simple computer game for the first 2-3 minutes of class that will lock up and not continue to work due to a programming error.

Real-world application/discussion: What if you were a brain surgeon and you were using a software application to help you do the surgery and it crashed?

PRESENTATION: (Teacher Activities)

- Introduce five different types of computer programming errors.
 - o System, Syntax, Semantic, Logic, and Runtime
- Teacher will lead students in correcting the simple programming error that was in the computer game (EvenOdd.java).

APPLICATION: (Student Activities)

- (Activity 1) Students will pair up in groups of two and will solve a computer programming error that is provided by the teacher (FahrenheitGUI.java)
- (Activity 2) Students will individually write up their own computer programs with simple programming errors and then trade with their partner and work on solving their partner's programming error.

ASSESSMENT/EVALUATION:

- Teacher observation during student Activity 1 and 2.
- Teacher evaluation using rubric for Activity 1 and 2.
- Ongoing assessment for debugging their computer programs throughout the school year.

CLOSURE:

- The teacher will recap the five different types of programming errors and start a classroom discussion on how these problem solving skills can be used in everyday life situations.

```
import javax.swing.JOptionPane;

//*****
// This program contains 2 errors that students must identify and correct.
//*****
public class EvenOdd
{
    // Determines if the number input by user is even or odd
    public static void main (String[] args)
    {
        String numStr, result;
        int num, again;

        do
        {
            //Displays an input dialog box. Returns a string.
            numStr = JOptionPane.showInputDialog("Enter an integer: ");
            num = Integer.parseInt(numStr);
            result = "That number is " + ((num%2 == 1) ? "even" : "odd");

            //Displays a message. First arg is parent. If null, then
            // box is centered on the screen. Second arg is message.
            JOptionPane.showMessageDialog(null, result);

            //Displays a message and Yes/No buttons. Returns an integer
            again = JOptionPane.showConfirmDialog (null, "Do another?");
        }
        while (again == JOptionPane.NO_OPTION);
    }
}
```

```

import javax.swing.JOptionPane;

//*****
// This program shows the usage of the 3 types of JOptionPane dialog boxes.
//*****
public class EvenOdd
{
    // Determines if the number input by user is even or odd
    public static void main (String[] args)
    {
        String numStr, result;
        int num, again;

        do
        {
            //Displays an input dialog box. Returns a string.
            numStr = JOptionPane.showInputDialog("Enter an integer: ");
            num = Integer.parseInt(numStr);
            result = "That number is " + ((num%2 == 0) ? "even" : "odd");

            //Displays a message. First arg is parent. If null, then
            // box is centered on the screen. Second arg is message.
            JOptionPane.showMessageDialog(null, result);

            //Displays a message and Yes/No buttons. Returns an integer
            again = JOptionPane.showConfirmDialog (null, "Do another?");
        }
        while (again == JOptionPane.YES_OPTION);
    }
}

```

```
//*****  
// Fahrenheit.java          Author:Lewis/Loftus/Cocking  
//  
// Demonstrates use of JFrame and JTextArea GUI components  
//*****  
  
public class Fahrenheit  
{  
    public static void main (String[] args)  
    {  
        FahrenheitGUI converter = new FahrenheitGUI();  
        converter.display();  
    }  
}
```

```

//*****
// FahrenheitGUI.java          Author: Lewis/Loftus/Cocking
//
// Demonstrates use of JFrame and JTextArea GUI components
//*****

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

public class FahrenheitGUI
{
    private int WIDTH = 300;
    private int HEIGHT = 75;

    private JFrame frame;
    private JPanel panel;
    private JLabel inputLabel, outputLabel, resultLabel;
    private JTextField fahrenheit;

    //*****
    // Sets up GUI
    //*****
    public FahrenheitGUI()
    {
        frame = new JFrame("Temperature Conversion");
        frame.setBounds(300, 300, 300, 300);
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        inputLabel = new JLabel ("Enter Fahrenheit temperature:");
        outputLabel = new JLabel("Temperature in Celsius: ");
        resultLabel = new JLabel("----")

        // Accept up to 5 chars in text field
        fahrenheit = new JTextField(5);
        fahrenheit.addActionListener (new TempListener());
        panel = new JPanel();
        panel.setPreferredSize (new Dimension(WIDTH, HEIGHT));
        panel.setBackground (Color.yellow);

        // Order in which labels and text field are added to the panel is
the order in which
        // they appear. Size of panel determines how components line up.
        panel.add (inputLabel);
        panel.add (fahrenheit);
        panel.add (outputLabel);
        panel.add (resultLabel);

        frame.getContentPane().add (panel);    //Add panel to frame
    }

    //*****
    // Display the frame
    //*****
    public void display()
    {
        frame.pack();    //Change size of frame to fit the panel
        frame.show();
    }
}

```

```
//*****  
// Represents action listener for the temperature input field  
//*****  
private class TempListener implements ActionListener  
{  
    public void actionPerformed (ActionEvent event)  
    {  
        int fahrenheitTemp, celsiusTem;  
        String text = fahrenheit.getText();  
        fahrenheitTemp = Integer.parseInt(text);  
        celsiusTemp = (fahrenheitTemp-32) * 5/9;  
        resultLabel.setText (Integer.toString(celsiusTemp));  
    }  
}  
}
```

```

//*****
// FahrenheitGUI.java          Author: Lewis/Loftus/Cocking
//
// Demonstrates use of JFrame and JTextArea GUI components
//*****

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

public class FahrenheitGUI
{
    private int WIDTH = 300;
    private int HEIGHT = 75;

    private JFrame frame;
    private JPanel panel;
    private JLabel inputLabel, outputLabel, resultLabel;
    private JTextField fahrenheit;

    //*****
    // Sets up GUI
    //*****
    public FahrenheitGUI()
    {
        frame = new JFrame("Temperature Conversion");
        frame.setBounds(300, 300, 300, 300);
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        inputLabel = new JLabel ("Enter Fahrenheit temperature:");
        outputLabel = new JLabel("Temperature in Celsius: ");
        resultLabel = new JLabel("---");

        // Accept up to 5 chars in text field
        fahrenheit = new JTextField(5);
        fahrenheit.addActionListener (new TempListener());
        panel = new JPanel();
        panel.setPreferredSize (new Dimension(WIDTH, HEIGHT));
        panel.setBackground (Color.yellow);

        // Order in which labels and text field are added to the panel is
the order in which
        // they appear. Size of panel determines how components line up.
        panel.add (inputLabel);
        panel.add (fahrenheit);
        panel.add (outputLabel);
        panel.add (resultLabel);

        frame.getContentPane().add (panel);    //Add panel to frame
    }

    //*****
    // Display the frame
    //*****
    public void display()
    {
        frame.pack();    //Change size of frame to fit the panel
        frame.show();
    }
}

```

```
//*****  
// Represents action listener for the temperature input field  
//*****  
private class TempListener implements ActionListener  
{  
    public void actionPerformed (ActionEvent event)  
    {  
        int fahrenheitTemp, celsiusTemp;  
        String text = fahrenheit.getText();  
        fahrenheitTemp = Integer.parseInt(text);  
        celsiusTemp = (fahrenheitTemp-32) * 5/9;  
        resultLabel.setText (Integer.toString(celsiusTemp));  
    }  
}  
}
```

Computer Programming

Name: _____

Period: _____ **Date:** _____

Grading Rubric: Debugging and Programming

0 = Student didn't identify program errors and didn't modify the computer program to fix them.

1 = Student identified program errors, but didn't modify the computer program to fix them.

2 = Student identified some of the program errors and modified the program to fix some of them.

3 = Student identified all of the program errors and modified the program to fix all of them.

Student Activity 1 grade: _____

Student Activity 2 grade: _____

Computer Programming

Name: _____

Period: _____ **Date:** _____

Grading Rubric: Debugging and Programming

0 = Student didn't identify program errors and didn't modify the computer program to fix them.

1 = Student identified program errors, but didn't modify the computer program to fix them.

2 = Student identified some of the program errors and modified the program to fix some of them.

3 = Student identified all of the program errors and modified the program to fix all of them.

Student Activity 1 grade: _____

Student Activity 2 grade: _____