CSUN | CALIFORNIA STATE UNIVERSITY NORTHRIDGE

COMP110L

COMP110

DR JEFF SOFTWARE
INDIE APP DEVELOPER

Intro to Algorithms & Programming

LAB

Part 1

Dr Jeff Drobman

Dr Jeff Software

website → *drjeffsoftware.com*

email → *jeffrey.drobman@csun.edu*

Index

CSUN
CALIFORNIA STATE UNIVERSITY NORTHRIDGE
COMP110L

Lab Programs

DR JEFF SOFTWARE
INDIE APP DEVELOPER

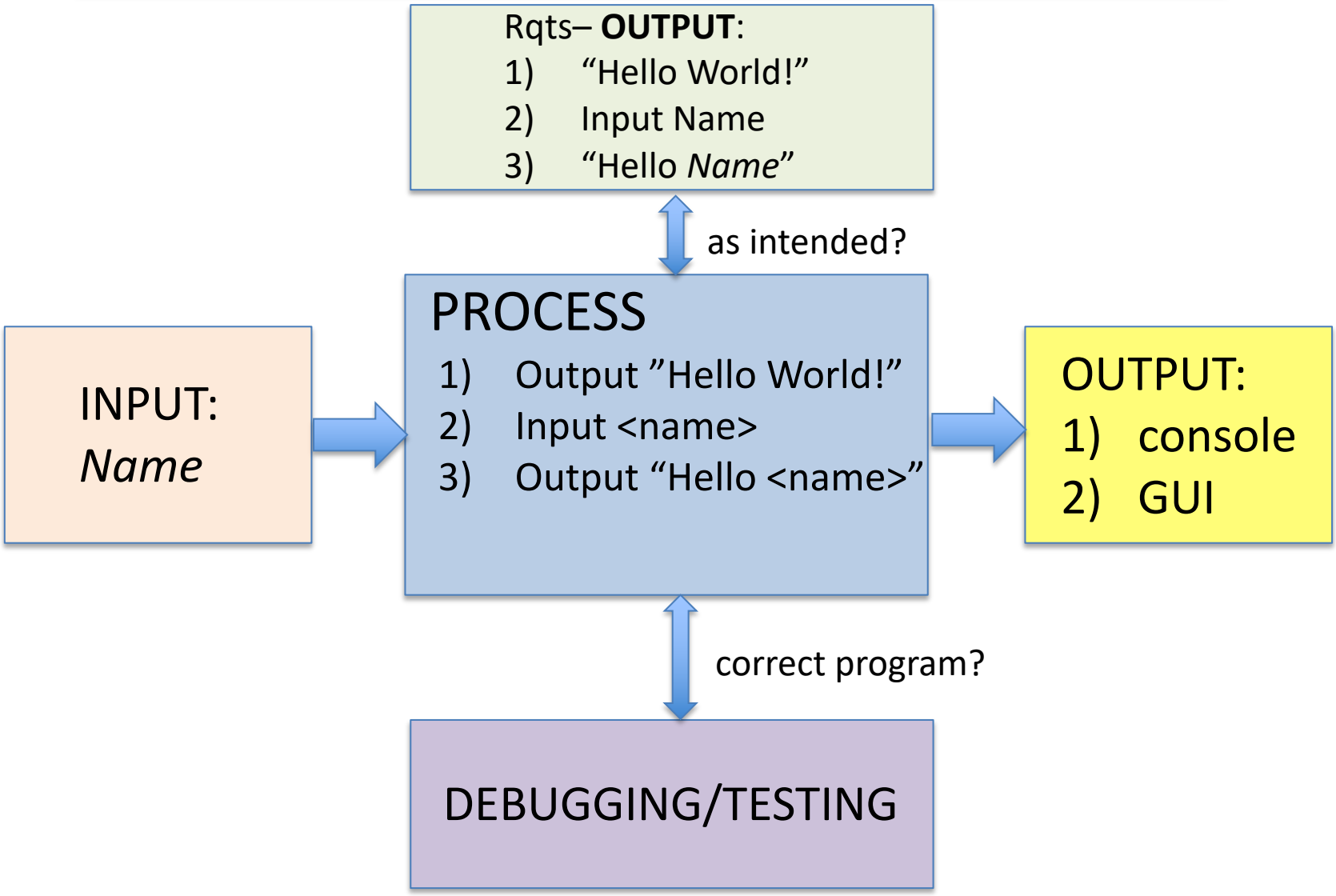
1. Hello World (I/O)
2. Temperature conversion (IF-THEN, *numerics, formatted* output)
3. Guess Secret Name (Input, IF-THEN, loops)
4. *Palindromes/Anagrams* (strings, *methods*)
5. *Homonyms* (strings, *methods,* arrays, *files*)
6. Prime numbers (algorithms, loops, *methods,* arrays, *files*)
7. *Cryptography/blockchains* (algorithms, *methods*)
8. Tic-Tac-Toe (arrays, *methods, formatted* output, *Classes*)
9. Bowling League (arrays, *files*, *methods*, stats, *Classes*)
10. Calendar (algorithms, *formatted* output, Date/Time)
11. Games (arrays, random numbers) → Project
12. Probability (*factorials-> recursion*)

Lab

LAB 1

Hello World

Lab 1: Hello World



Comparison: "Hello World"

Lab 1

C

```c
#include <stdio.h>
int main (void) {
  printf("Hello world!\n");
}
```

C++

```cpp
#include <iostream>
int main () {
  std::cout << "Hello world!\n";
}
```

Java

```java
public class helloWorld {
public static void main (String[] args) {
  system.out.println ("Hello world!");
} }
```

Javascript

```javascript
//myfile.js
Console.log("Hello world!");
```

Python

```python
Print "Hello world!"
```

Comparison: "Hello World"

Basic

```
10 PRINT "Hello, world!"
20 END
```

note: line numbers!

VB

```
Public Sub Main()
  MsgBox "Hello, world!"
End Sub
```

OOP + **GUI**

C#

```
using System;

internal static class HelloWorld
{
    private static void Main()
    {
        Console.WriteLine("Hello, world!");
    }
}
```

OOP + console

DOS

```
@echo Hello World!
```

script (for console)

Comparison: "Hello World"

PHP

```php
<?php
print "Hello world!";
?>
```

- all console

Assembly

```
.model small
.stack 100h

.data
msg     db      'Hello world!$'

.code
start:
        mov     ah, 09h
        lea     dx, msg
        int     21h
        mov     ax, 4C00h ;
        int     21h
end start
```

Hello World – Console

Java

header

main

console out

```
1  /* CSUN CS110 header
2  student:
3  date:
4  file:  Hello.java
5  Lab 1:  Hello
6  */
7  //main class
8  public class Hello {
9  //main method
10 public static void main(String[] args) {
11    //code starts here
12    System.out.println("Hello World!");
13
14 } //end main method
15 //end class
16 }
```

CSUN
CALIFORNIA STATE UNIVERSITY NORTHRIDGE
COMP110L

Hello World – Console

DR JEFF SOFTWARE
INDIE APP DEVELOPER

zyLabs | Lab 1

Ch 1 Warm up: Hello world (Java)

This zyLab activity prepares a student for a full programming assignment.

For each of the following steps, end the program's output with a newline.

(1) Write a program that outputs the following. (1 pt)

```
Hello world!
```

(2) Update to output the following. (1 pt)

```
Hello world!
How are you?
```

(3) Finally, update to output the following. (1 pt)

```
Hello world!
How are you?
   (I'm fine).
```

LAB ACTIVITY | 20.4.1: Ch 1 Warm up: Hello world (Java)

CSUN CALIFORNIA STATE UNIVERSITY NORTHRIDGE

COMP110L

Hello World – Console

DR JEFF SOFTWARE INDIE APP DEVELOPER

zyLabs Lab 1

Main.java

Load default template...

```
1  /*student:
2  date:
3  Lab 1:  <name>
4  */
5
6  import java.util.Scanner;
7
8  public class Main {
9     public static void main(String[] args) {
10       //Scanner scnr = new Scanner(System.in);
11       System.out.println("Hello World!");
12       /* Type your code here. */
13    }
14 }
15
```

- I supply "starter" source code here
- You supply YOUR source code here

Develop mode | Submit mode

Run your program as often as you'd like, before submitting for grading. Below, type any needed input values in the first box, then click **Run program** and observe the program's output in the second box.

Enter program input (optional)

If your code requires input values, provide them here.

I P O

Run program

Input (from above) → Main.java (Your program) → Output (shown below)

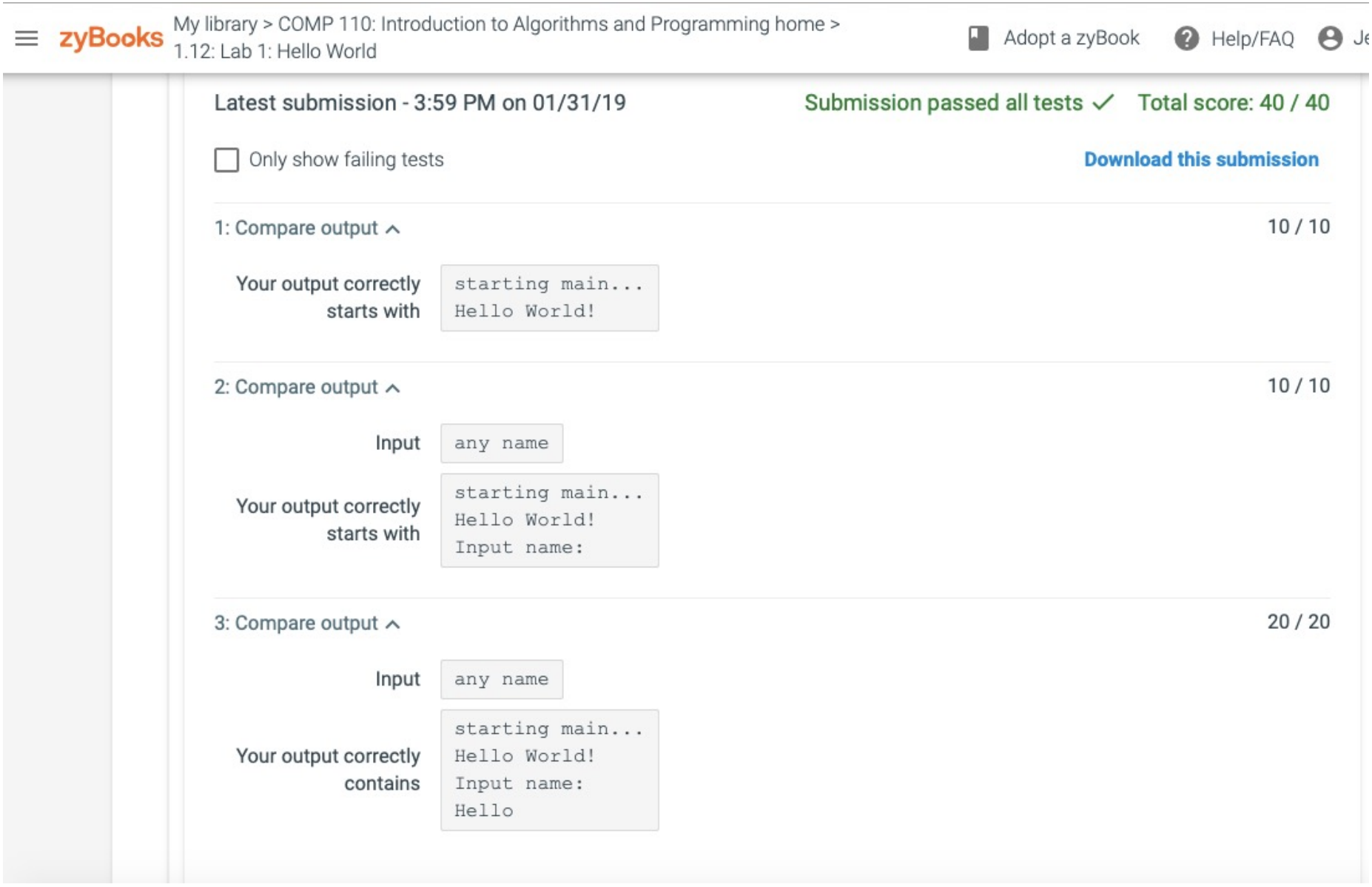
Program output displayed here

```
Hello World!
```

CSUN
CALIFORNIA STATE UNIVERSITY NORTHRIDGE
COMP110L

zyLab 1 Tests

DR JEFF SOFTWARE
INDIE APP DEVELOPER



CSUN
CALIFORNIA STATE UNIVERSITY NORTHRIDGE
COMP110L

GUI: "Hello World"

DR JEFF SOFTWARE
INDIE APP DEVELOPER

Lab 1

Java

```
//add ref to GUI class
import javax.swing.JOptionPane;   -OR-   import javax.swing.*;
//main code
public class helloWorld {
public static void main (String[] args) {
 JOptionPane.showMessageDialog(null, "Hello world!");
}
}
```

javax.swing.JOptionPane.showMessageDialog

- GUI

CSUN
CALIFORNIA STATE UNIVERSITY NORTHRIDGE

COMP110L

Hello World – Combined

DR JEFF SOFTWARE
INDIE APP DEVELOPER

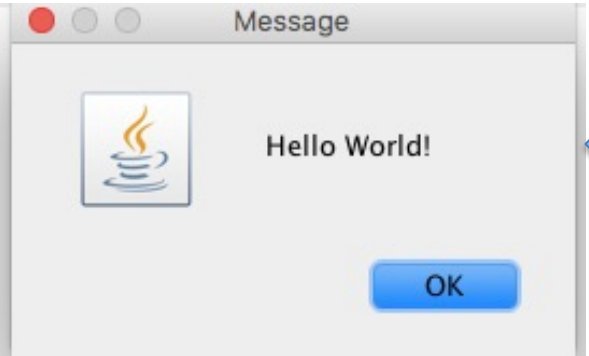
Lab 1

```
1    /* CSUN CS110 header
2    student:
3    date:
4    file:  Hello.java
5    Lab 1:  Hello
6    */
7    //imports
8    import javax.swing.*;
9    //main class
10   public class Hello {
11   //main method
12     public static void main(String[] args) {
13     //code starts here
14       System.out.println("Hello World!");
15       JOptionPane.showMessageDialog(null, "Hello World!");
16     } //end main method
17   } //end class
```

➢ GUI

```
----jGRASP exec: java Hello
Hello World!
```

➢ Console



Mac version

Hello World + Input


```
 7 //imports
 8 import java.util.*;
 9 import javax.swing.*;
10 //main class
11 public class Lab1Hello {
12 //main method
13    public static void main(String[] args) {
14       //code starts here (indent)
15       //OUTPUT
16       System.out.println("Hello World!");//console
17       JOptionPane.showMessageDialog(null, "Hello World!");//GUI
18       //INPUT
19       Scanner input = new Scanner(System.in);//instantiate "Scanner" class
20       System.out.print("Input name: ");//prompt
21       String name = input.nextLine();
22       String msg = "Hello " + name;//print msg
23       //OUTPUT again
24       System.out.println(msg);//console
25       JOptionPane.showMessageDialog(null, msg);//GUI
26    } //end main method
27 } //end class
```

```
----jGRASP exec: java Lab1Hello
Hello World!
Input name: Jeff How D
Hello Jeff How D

----jGRASP: operation complete.
```

Hello World – Plus

- Console
- Math extra

Lab 1

```
3  date:
4  file:  HelloPlus.java
5  Lab 1:  Hello
6  */
7  //main class
8  public class HelloPlus {
9  //main method
10 public static void main(String[] args) {
11    //code starts here
12    System.out.println("Hello Math!");
13    //Integers
14    int x = 3, y = 7;
15    System.out.println("x-y=" + (x-y));
16    System.out.println("x/y=" + (x/y) + " >quotient");
17    System.out.println("x%y=" + (x%y) + " >remainder");
18    //Floating point
19    float xx = 3, yy = 7;
20    System.out.println("x-y=" + (xx-yy));
21    System.out.println("x/y=" + (xx/yy) + " >quotient");
22    System.out.println("x%y=" + (xx%yy) + " >remainder");
23 } //end main method
24 }//end class
```

```
----jGRASP exec: java HelloPlus
Hello Math!
x-y=-4
x/y=0 >quotient
x%y=3 >remainder
x-y=-4.0
x/y=0.42857143 >quotient
x%y=3.0 >remainder

----jGRASP: operation complete.
```

Lab 1 Form

Lab 1

Requirements
1. Print “Hello World” on both console and GUI box
2. Input (console) your name
3. Print “Hello <your name>” on both console and GUI box

Inputs

- Console

```
Dr Jeff
```

- GUI

Outputs

```
----jGRASP exec: java Lab1Hello
Hello World!
Dr Jeff
Hello Dr Jeff

----jGRASP: operation complete.
Hello Math!
x-y=-4
x/y=0 >quotient
x%y=3 >remainder
```

Lab

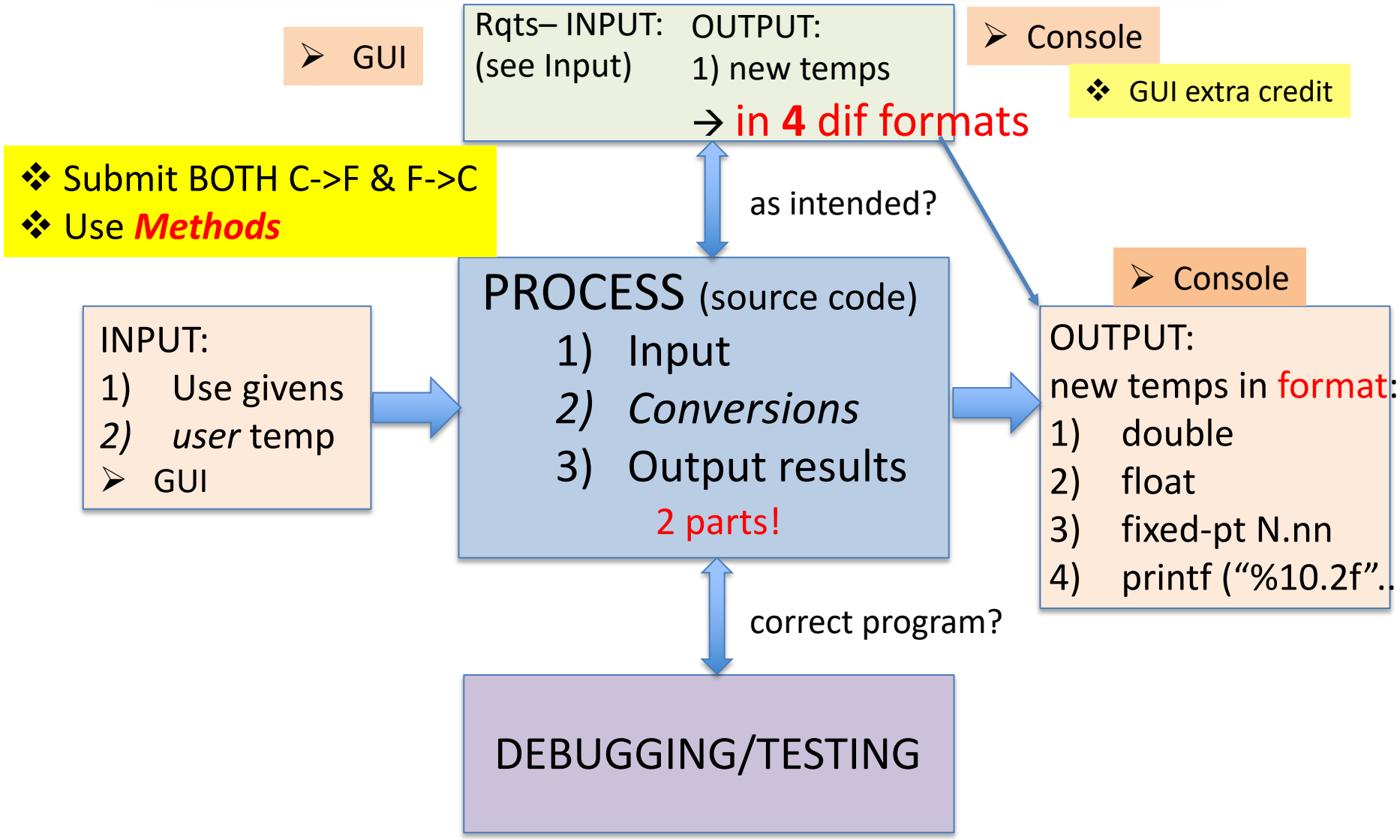
LAB 2

Temp Conversion
& Formatting

CSUN
CALIFORNIA STATE UNIVERSITY NORTHRIDGE
COMP110L

Lab 2: Temp Conv

DR JEFF SOFTWARE
INDIE APP DEVELOPER



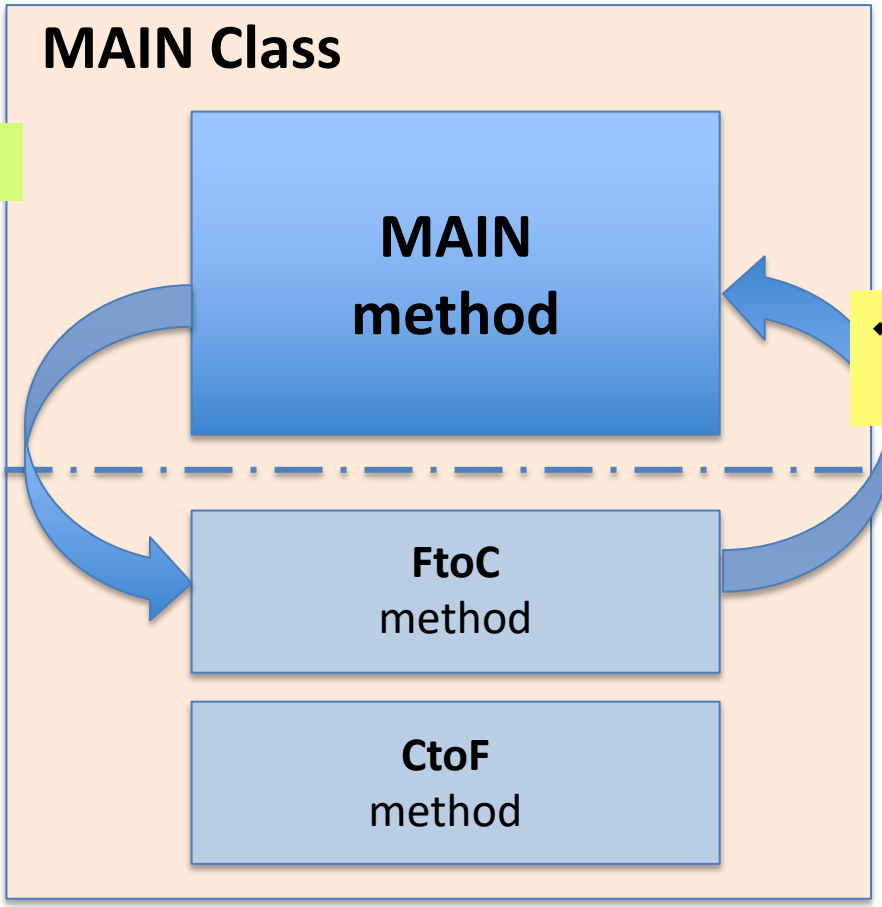
Structure (Macro)

Lab 2

OOP
Structures

Classes/methods

Execution is by *call* sequence



MAIN *Class* = any name

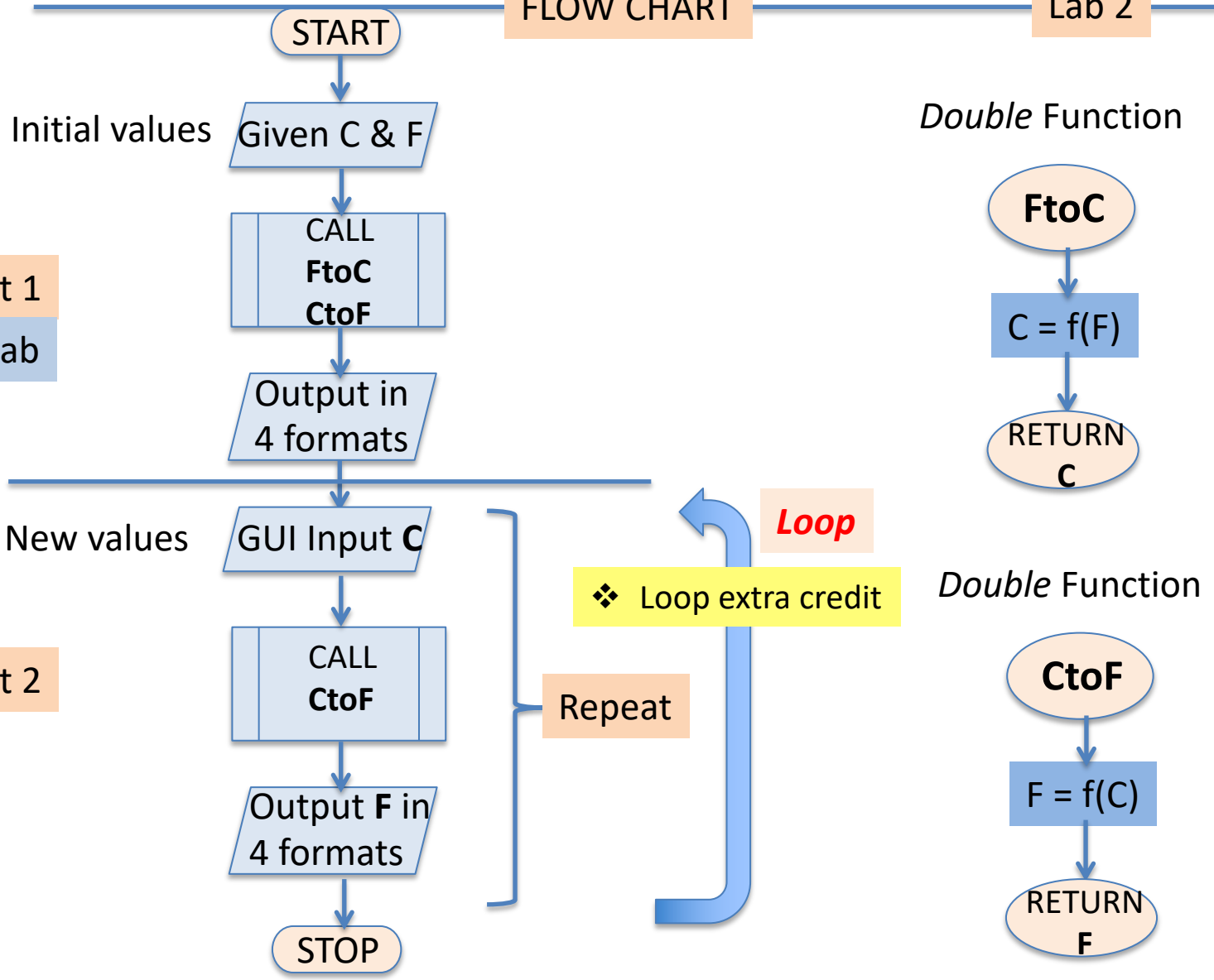
MAIN *Method* = "main"

❖ Place "main" method FIRST

Minimum required

Flow Chart

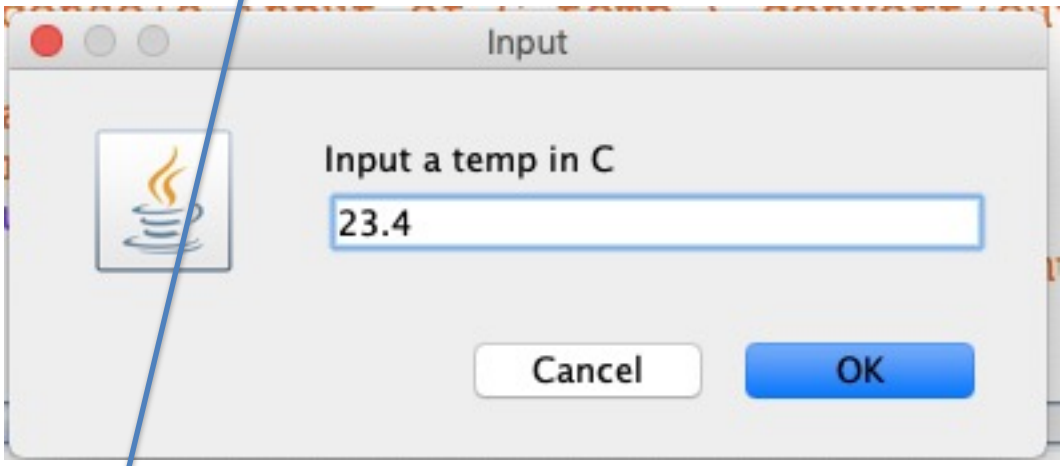
FLOW CHART — Lab 2



Part 2: GUI Input of C

Lab 2

```
41      //GUI:  input + test (with extra args)
42      String inStrC = JOptionPane.showInputDialog("Input a temp in C");
43      double guiC = Double.parseDouble(inStrC);//convert Str->double
44      double guiF = CtoF(guiC);
45      System.out.printf("===\n %.2f %s %.2f %s", guiC, "in C = ", guiF, "in F");
46      JOptionPane.showMessageDialog(null,String.format("%10.2f",dblC),"Test",3);
```




```
===
  23.40 in C =  74.12 in F
  ----jGRASP: operation complete.
```

console output

CSUN
CALIFORNIA STATE UNIVERSITY NORTHRIDGE
COMP110L

GUI Input Extended

DR JEFF SOFTWARE
INDIE APP DEVELOPER

Lab 2

```
String input = JOptionPane.showInputDialog("enter input: ");
```

```
xx = Integer.parseInt(ss);
```

can combine

```
temp = Double.parseDouble(JOptionPane.showInputDialog("Enter the temperature in Celsius: "));
```

check type

```
if (isInt) {
int xint = Integer.parseInt(inX.trim());
```

Temp Conversion

Lab 2

$$F = C * (9/5) + 32$$
$$C = (F-32) * (5/9)$$

- (9/5) = 1.8 ⟸ exact value
- (5/9) = 0.555… ⟸ repeating decimal

❖ What you should learn

→how to represent a fixed-point number (literals vs. vars)
→how to use mixed types in expressions
→how to truncate (and round) extra digits
→how to use formatted output

- Double > Float > Long > Int > Short > Byte

CSUN CALIFORNIA STATE UNIVERSITY NORTHRIDGE
COMP110L

Lab: Type Conversions

DR JEFF SOFTWARE INDIE APP DEVELOPER

Lab 2

- Java *truncates* Integers
  - to *Round* add 0.5

5/9 → 0
5/9 + 0.5 → 1.0555 → 1

- Expressions
  - mixed types resolve to highest precision operand
    - Double > Float > Long > Int > Short > Byte

5.0/9 → 0.5555
5/9 → 0

- Casting
  - Implicit

int i = 1.23 → 1
int i = 1.23e+12 → *error*
float x = 1.23 → 1.23
byte x = 128 → *error*

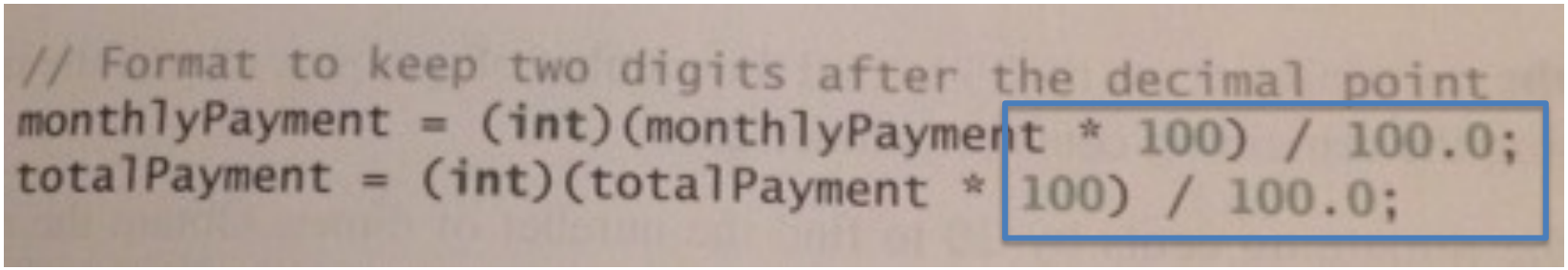
  - Explicit

float f = 1.23 → 1.23e0
int i = f + 1.23 → *error?*
int i = (int) f → 1
int i = (int) 1.23e+12 → *error*
long i = (int) 1.23e+12 → 1,230,000,000,000

```
float fahrenheit = celsius * (9.0f / 5f) + 32f;
JOptionPane.showMessageDialog(null, celsius+ " is " + fahrenheit + "
```

Formatted Precision

123.45



```
// Format to keep two digits after the decimal point
monthlyPayment = (int)(monthlyPayment * 100) / 100.0;
totalPayment = (int)(totalPayment * 100) / 100.0;
```

(int) (x * 100) / 100.**0**

123.456

(int) (x * 1000) / 1000.**0**

Formatted Output

- Console

Lab 2

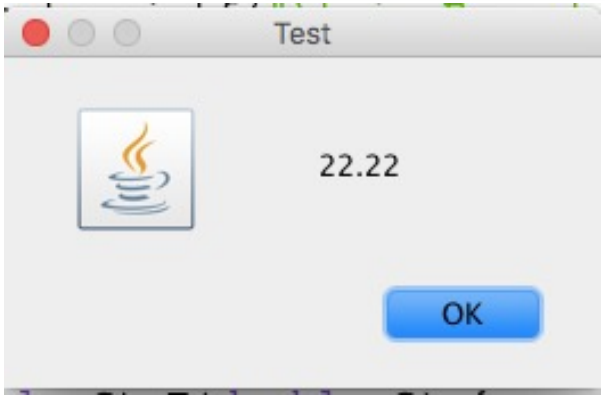
```
printf Method
.
System.out.printf("%b %c %d %f %e %s",
  true, 'A', 45, 45.5, 45.5, "Welcome");
System.out.printf("%-5d %10.2f %10.2e %8s",
  45, 45.5, 45.5, "Welcome");
```

- GUI

String.format(%10.2f, variable)

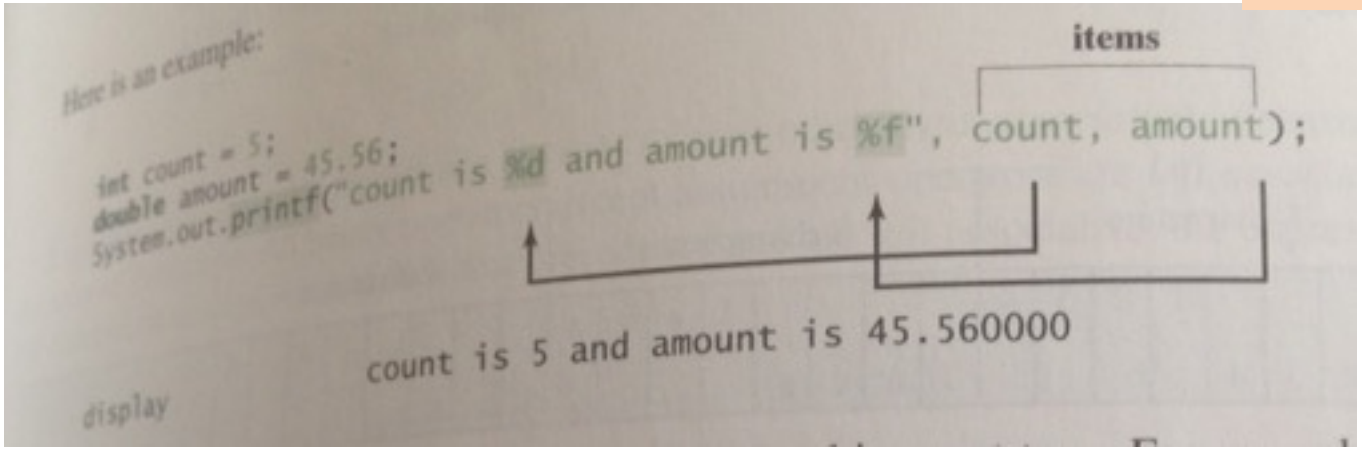
Sec 10.10.7
p. 390

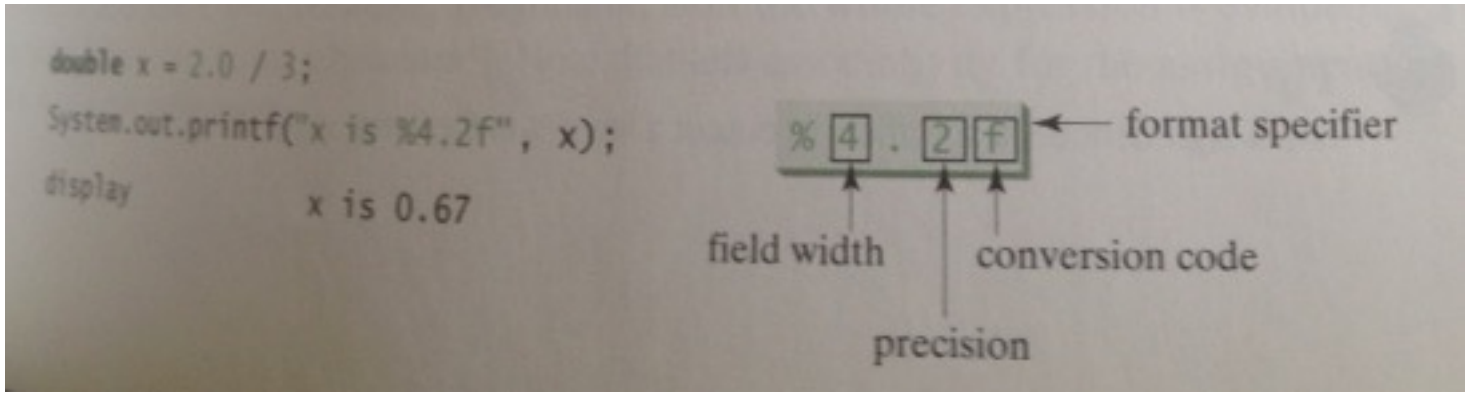
```
JOptionPane.showMessageDialog(null,String.format("%10.2f",dblC),"Test",3);
```



Formatted Output

Lab 2





Methods in Java


```
public static void main(String[ ], args)  {
    <statements>
    if ($F2C) tempC = FtoC(fahr);
    else tempF = CtoF(celc);
}
```

➢ CALL methods

```
public static double FtoC(double ftemp)  {
    double ctemp = (ftemp-32) *5/9.0;
    return ctemp;
}
```

```
public static double CtoF(double ctemp)  {
    double ftemp = ctemp *9/5.0 + 32;
    return ftemp;
}
```

- *Signature*
- Call
- Parameter passing
  - By ***Value***
- Return
  - Sub -> *void*
  - Function -> *value*

```
//start other methods
    static double FtoC(double F) {
        double C = 5./9. * (F - 32); //correct formula
        return C;
    }
    static double CtoF(double C) {
        double F = 1.8 * C + 32; //correct formula
        return F;
    }
```

Preferred Output

➢ Console

Lab 2

```
debug: starting code
Convert C to F:
25.123456 degC= 77.2222208 degF in Double
    in Float:  77.22222
    in Fixed:  77.0
    in Format:      77.22,    7.72e+01
Convert F to C:
72.0 degF= 22.2222222222222 degC in Double
    in Float:  22.222221
    in Fixed:  22.0
    in Format:      22.22,    2.22e+01

  ----jGRASP: operation complete.
```

fix this

extra

fix

extra

➢ Inputs

Inputs

```
Input Celsius temperature
25.123456
Input Fahrenheit temperature
72
```

Code – Error/Fix

Lab 2

```
//call conversion methods
double newC = FtoC(f);
double newF = CtoF(c);
System.out.println(c +" degC= " +newF +" degF in Double");
System.out.println(f +" degF= " +newC +" degC in Double");
float fltC = FtoC(f);
float fltF = CtoF(c);
```

fix

```
float fltC = (float) dblC;
float fltF = (float) dblF;
```

```
Lab3.java:23: error: incompatible types: possible lossy conversion
      float fltC = FtoC(f);
                   ^
Lab3.java:24: error: incompatible types: possible lossy conversion
      float fltF = CtoF(c);
```

Code – Complete

➢ Console

Lab 2

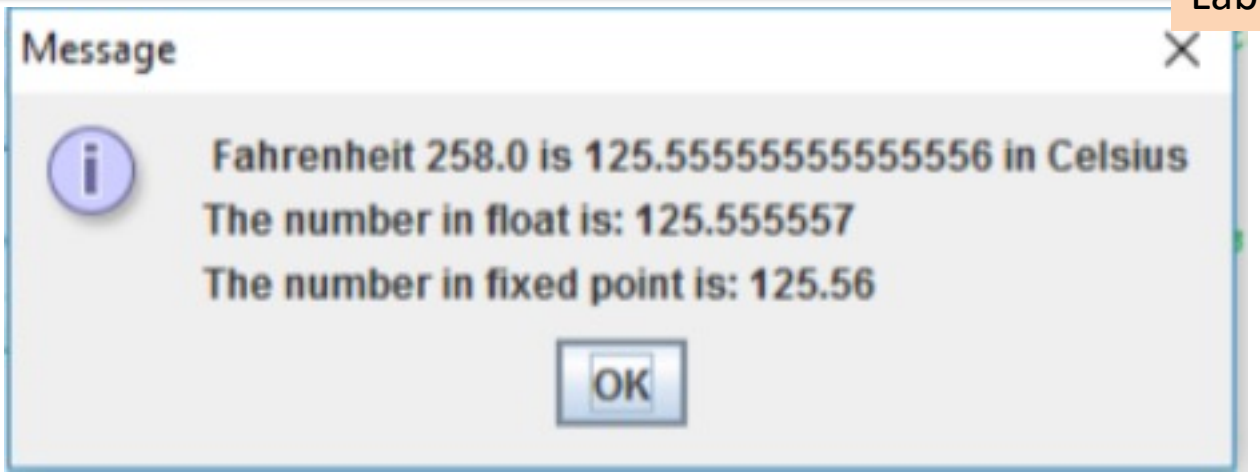
```
10 public class Lab3 {
11    static final boolean $DEBUG = true;
12 //main method
13    public static void main(String[] args) { //static?
14       if ($DEBUG) System.out.println("debug: starting code");
15       //code starts here: input data
16       double c = 25.123456, f = 72; //replace with Console Input!
17       //call conversion methods
18       double dblC = FtoC(f);
19       double dblF = CtoF(c);
20       float fltC = (float) dblC;
21       float fltF = (float) dblF;
22       float fixedF = (int) fltF;//sub correct code!
23       float fixedC = (int) fltC;//ditto
24       System.out.println("Convert C to F:");
25       System.out.println(c +" degC= " +dblF +" degF in Double");
26       System.out.println("\t in Float:  " +fltF);
27       System.out.println("\t in Fixed:  " +fixedF);
28       System.out.printf("\t in Format:  %10.2f, %10.2e\n", dblF, dblF);
29       System.out.println("Convert F to C:");
30       System.out.println(f +" degF= " +dblC +" degC in Double");
31       System.out.println("\t in Float:  " +fltC);
32       System.out.println("\t in Fixed:  " +fixedC);
33       System.out.printf("\t in Format:  %10.2f, %10.2e\n", dblC, dblC);
34    } //end main method
```

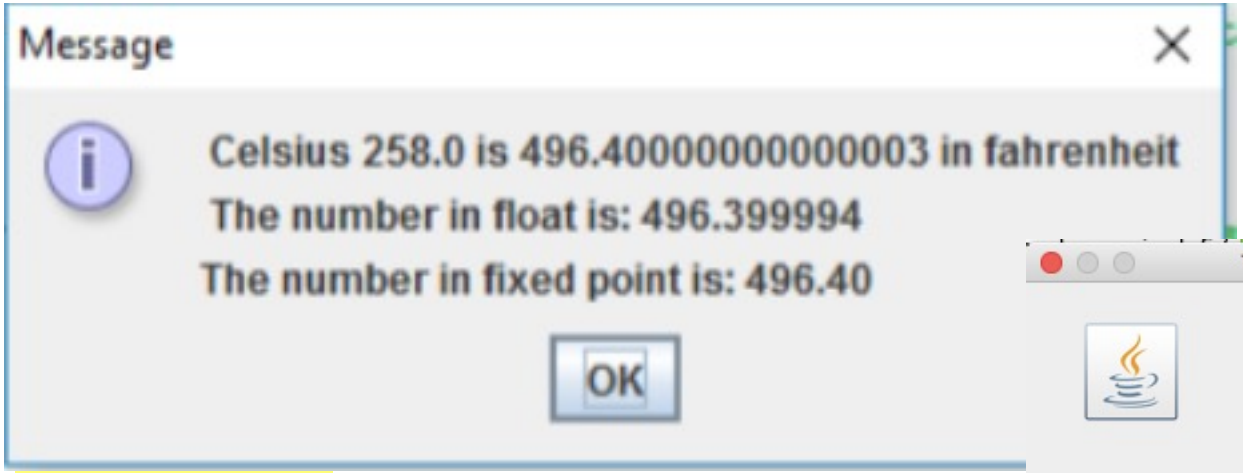
call methods

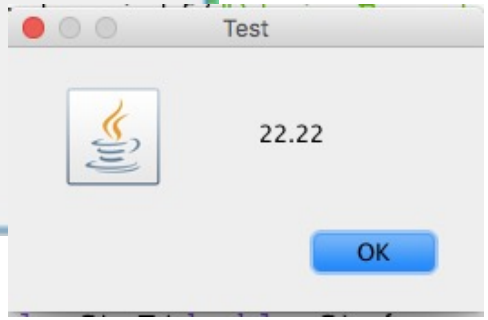
convert types

Example GUI Output

Lab 2







- Add GUI

String.format(%10.2f, variable)

```
JOptionPane.showMessageDialog(null,String.format("%10.2f",dblC),"Test",3);
//end main method
```

Lab

LAB 3

Secret Name Game

CSUN
CALIFORNIA STATE UNIVERSITY NORTHRIDGE
COMP110L

Lab 3: Secret Name

DR JEFF SOFTWARE
INDIE APP DEVELOPER

3 guesses

I/O:
1) admin: **either**
2) user: **GUI**

Rqts– INPUT: (see Input)
OUTPUT: (see Output)

- GUI

Part 1: 3 guesses-–ZyLab
Part 2: play again?—jGRASP

1) console
2) GUI

as intended?

PROCESS (source code)

- Input secret name
- Player select name*

Loop (<=3)
1) Input guess
2) Output results
3) Exit loop if **WIN**
4) Ask "Play again?"*

*Part 2

INPUT:
1) admin: secret name
2) user: select player
3) user: guesses

- ***Next player*** select from ***GUI Box***

OUTPUT:
1) correct–Win
2) not–# guesses
3) 0 guesses–Lose

correct program?

DEBUGGING/TESTING

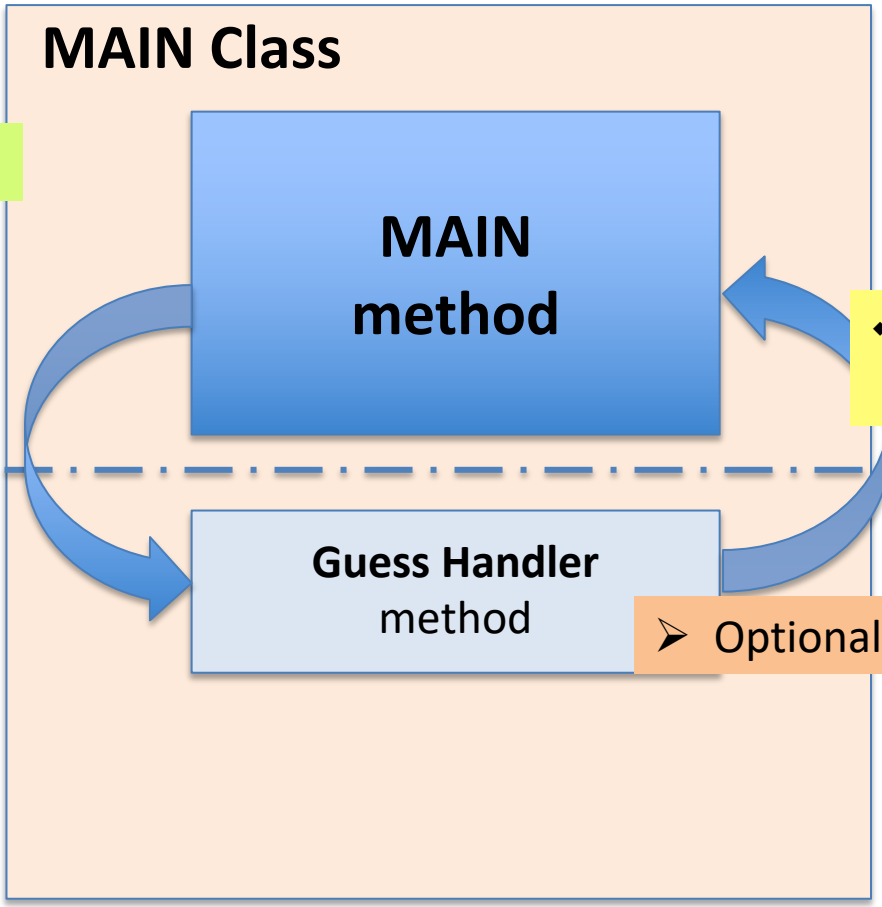
Structure (Macro)

Lab 3

OOP
Structures

Classes/methods

Execution is by *call* sequence



MAIN *Class* = any name

MAIN *Method* = "main"

❖ Place "main" method FIRST

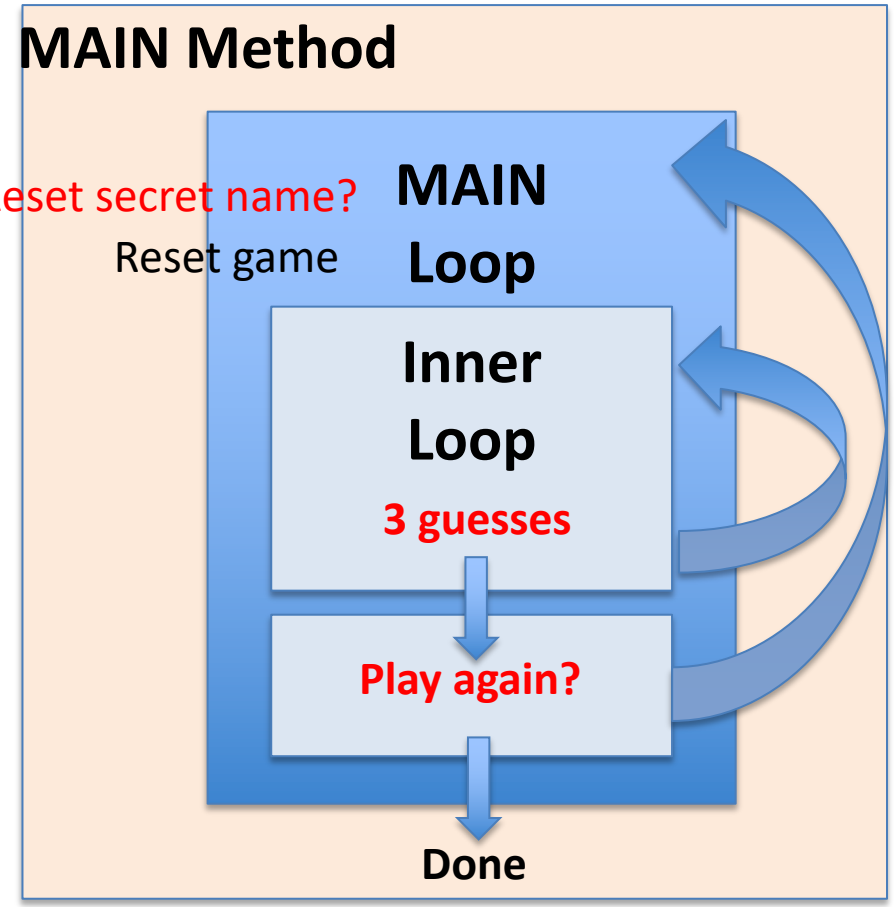
Minimum required

➢ Optional

Structure (Loops)

Lab 3

Loop Structures



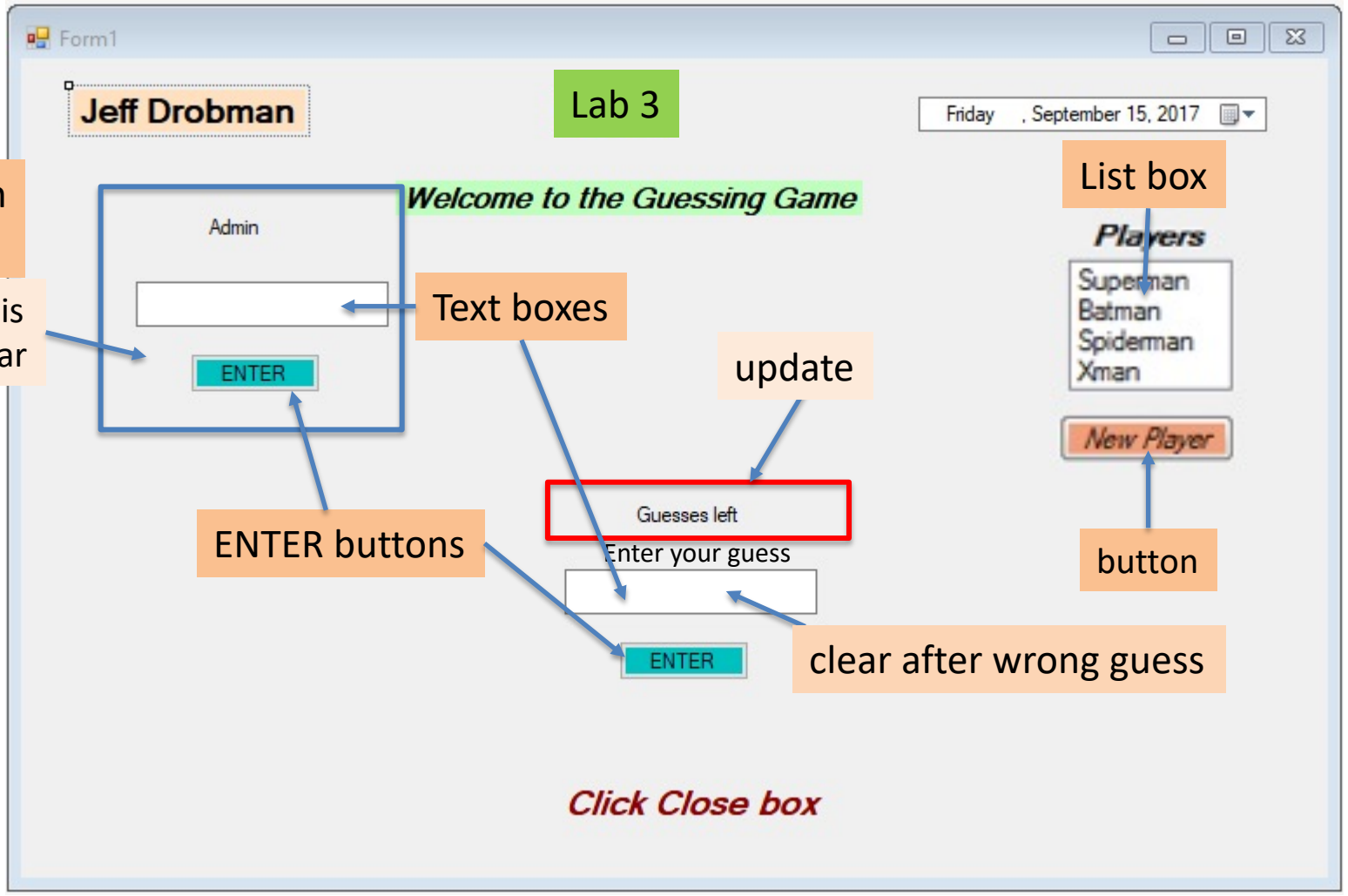
Outer While Loop

CSUN
CALIFORNIA STATE UNIVERSITY NORTHRIDGE
COMP110L

Lab 3 Form View

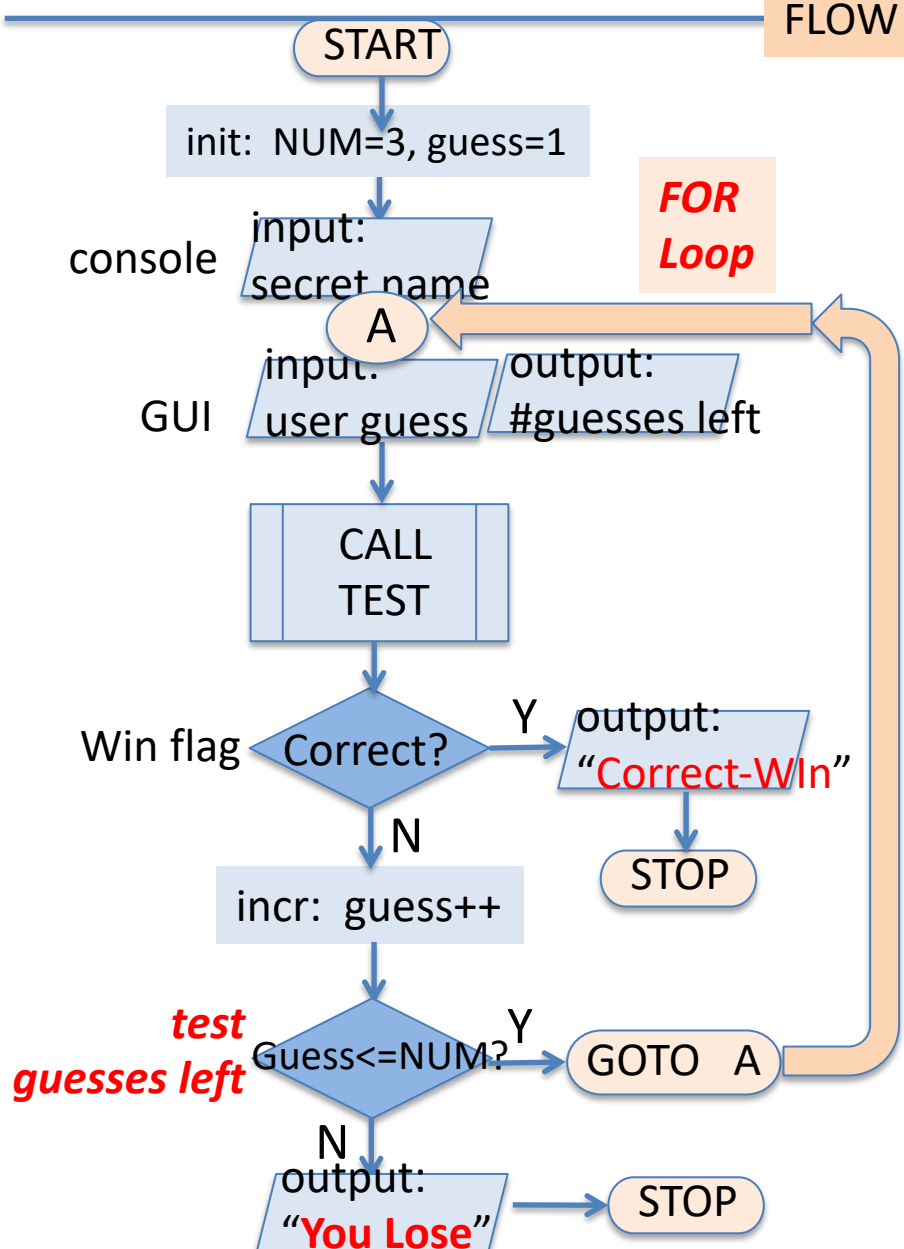
DR JEFF SOFTWARE
INDIE APP DEVELOPER

Lab 3

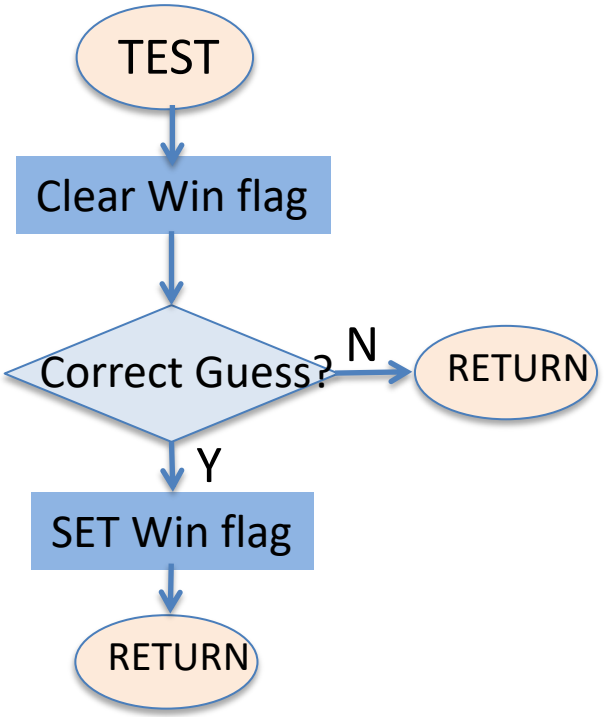


Lab 3: Guess Secret Name

FLOW CHART — Lab 3 — ***3 guesses*** — WIN or LOSE



Boolean Function



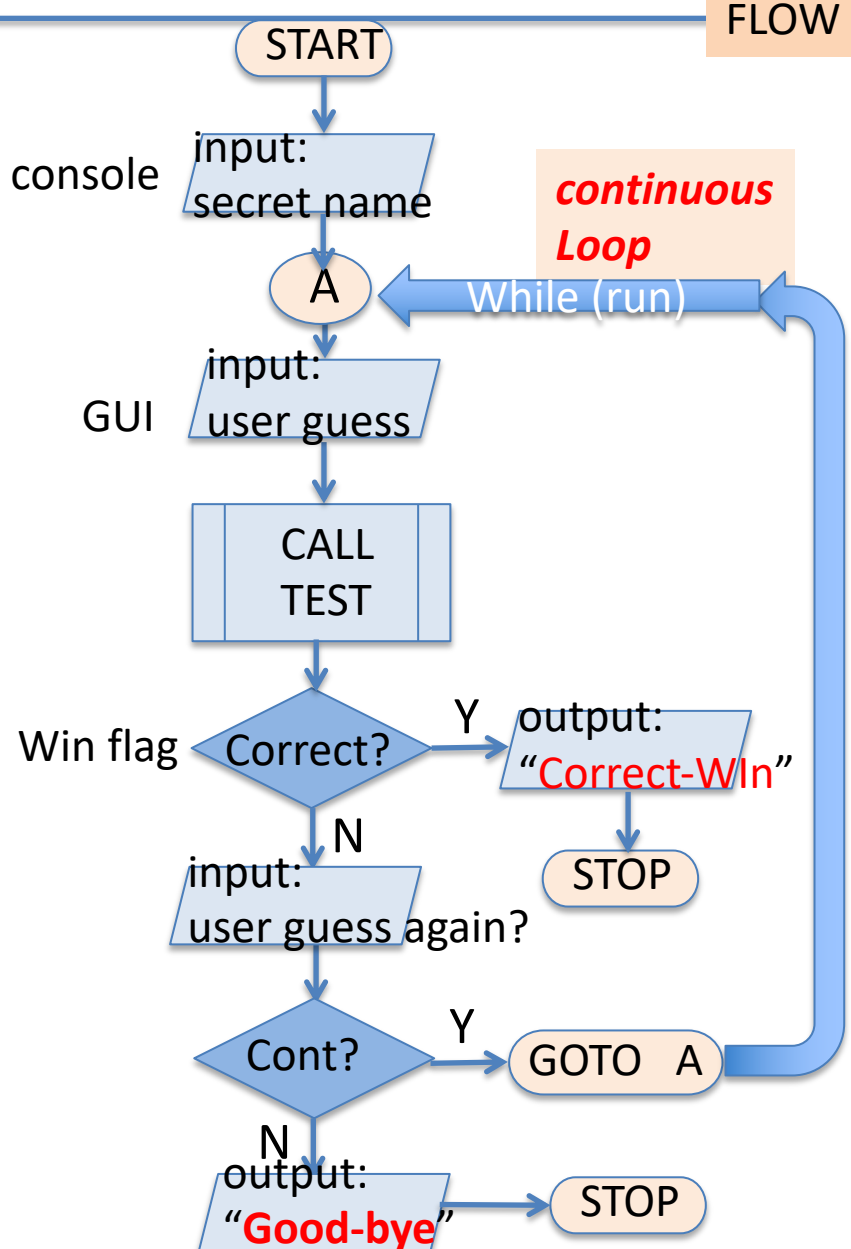
CSUN
CALIFORNIA STATE UNIVERSITY NORTHRIDGE
COMP110L

Lab 3: Guess Secret Name

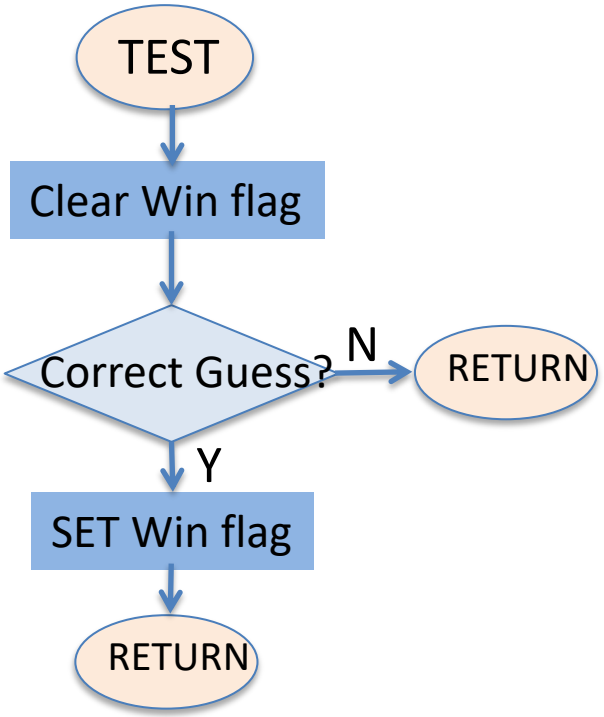
DR JEFF SOFTWARE
INDIE APP DEVELOPER

FLOW CHART | Lab 3 | ***Unlimited guesses***

WIN or Quit (no LOSE)



Boolean Function

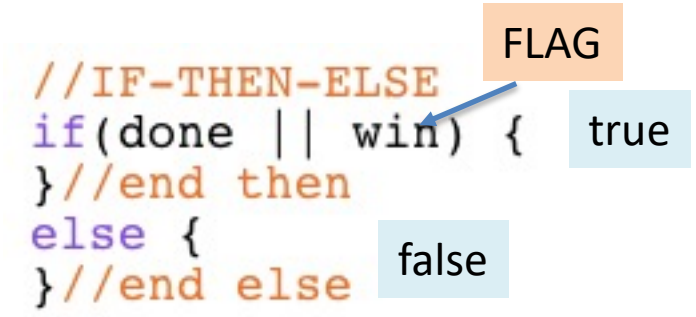


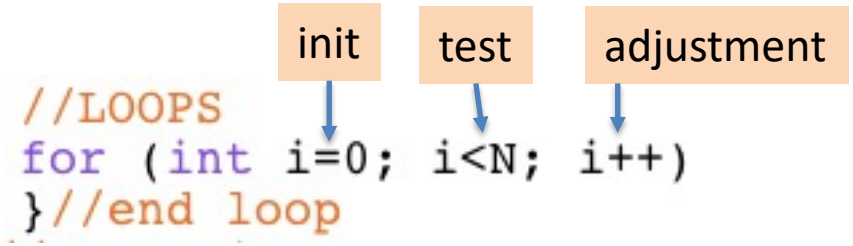
Stop vs. Break

- Break Loop
  - Either For or While loops
  - break;
  - continue;
- Stop
  - Program reaches end
  - System.exit(0);

Control Constructs

Lab 3





- 3 cases
  1) **Win**
  2) **Lose** (3 guesses only)
  3) Neither → continue

Key Variables

Inits

Lab 3

```
1  /* Dr Jeff Drobman test code
2  CSUN class CS110
3  file:  Lab2P1.java
4  */
5  import java.util.*;
6  import javax.swing.*;
7  //import java.io.*;
8
9  public class Lab2P1 {
10    static final boolean $DEBUG = true;
11    static String outMsg;
12    static final String empty = "";
13 public static void main(String[] args) {
14     boolean win = false, lose = false;
15     boolean done = false, invalid = false;
16     int count = 0, guesses = 3;
17     if($DEBUG) System.out.println("starting main...");
18     // **add code:  console input secret name
19     //start "for" loop
```

Global Strings

"Flags"

Counts

Enter Guesses: Loops


```
  //start "while" loop
while(!done) {//continue until done
```

Unlimited *guesses*

```
//**player guesses:  start "for" loop
  for(int i=0; i<numGuesses; i++) {
     String guessString = "You have " + leftGuesses + " guesses left.\n
     leftGuesses--;
     count++;
  //popup box:  "guess the secret name"
     String guessName = JOptionPane.showInputDialog(null,guessString);
     System.out.println("guess#" +count +"=" +guessName);//log on console
```

3 *guesses*

Handle Guess

- String.equalsIgnoreCase

```
// **add code:  test if guess correct; set "win"
   win = guessName.equalsIgnoreCase(secretName);//key test
//3 cases:  win, incorrect, lose
   if (win) outMsg = "You WON! \n Good-bye";//WIN
   else if (leftGuesses>0) outMsg = "Incorrect!";//not done
   else {//LOSE
      outMsg = "Incorrect!\nYou LOST!\nGood-bye!";
      lose = true;}
   JOptionPane.showMessageDialog(null, outMsg);
   if (win || lose)
      break;//stop loop
}//end loop
```

Also System.out.println(**outMsg**);

```
 ----jGRASP exec: java Lab3
starting main...
Now playing: Batman
guess#1=bob
You WON!
 Good-bye
```

I/O: Guesses

Mac version Lab 3

- GUI only on jGRASP!
- Replace with *Console* input on zyLab

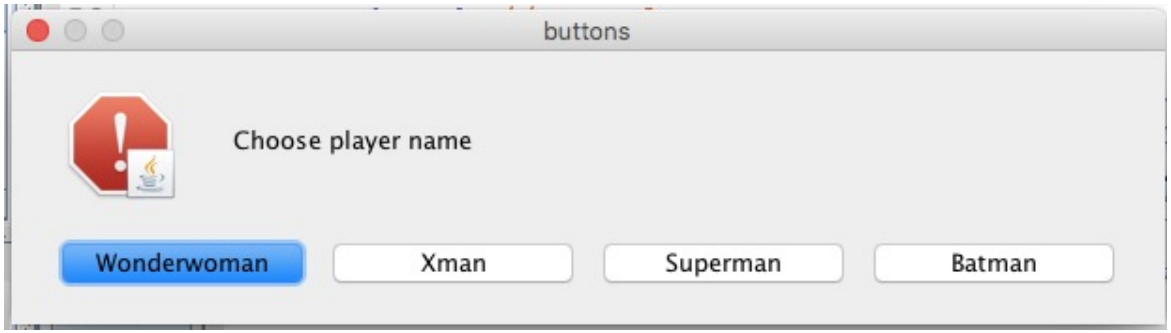
CSUN
CALIFORNIA STATE UNIVERSITY NORTHRIDGE
COMP110L

Enter Player Name

DR JEFF SOFTWARE
INDIE APP DEVELOPER

Lab 3 Part 2

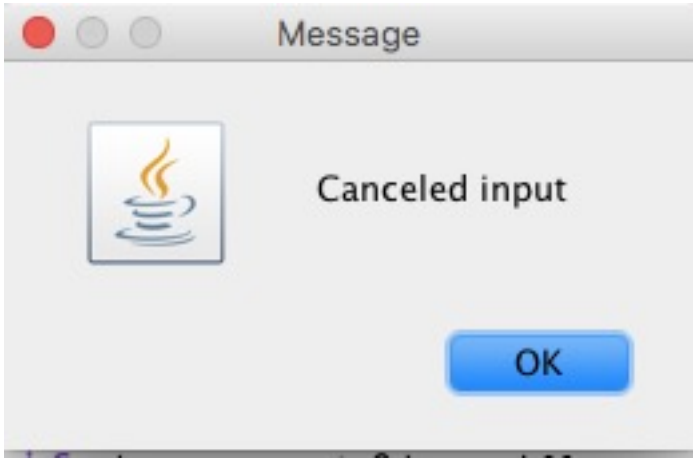
```
//**player: choose avatar (name)
  String[] avatars = {"Batman", "Superman", "Xman", "Wonderwoman"};
  int avNum = JOptionPane.showOptionDialog(null, "Choose player name",
  "buttons", 0, 0, null, avatars, avatars[3]);
  String playerName = avatars[avNum];//convert # to name
  JOptionPane.showMessageDialog(null, "Welcome " + playerName +"!");
  System.out.println("Now playing: " + playerName);//log
```

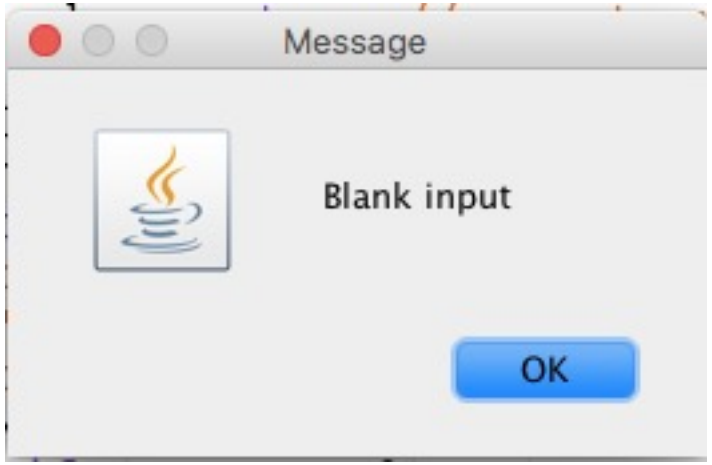


Mac version

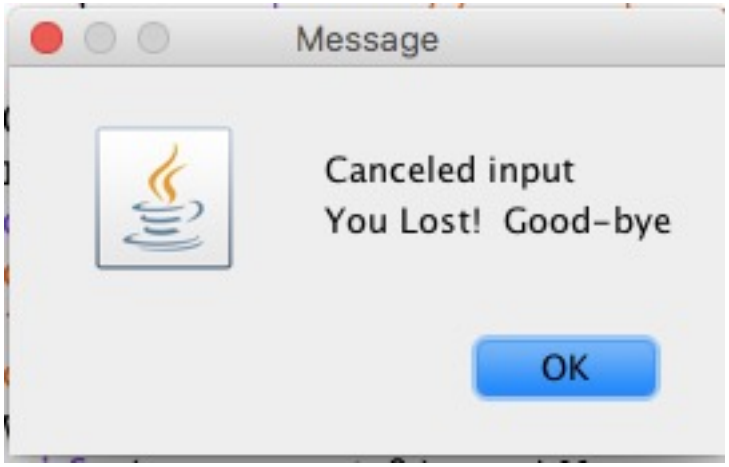
Lab 3 Extra Outputs

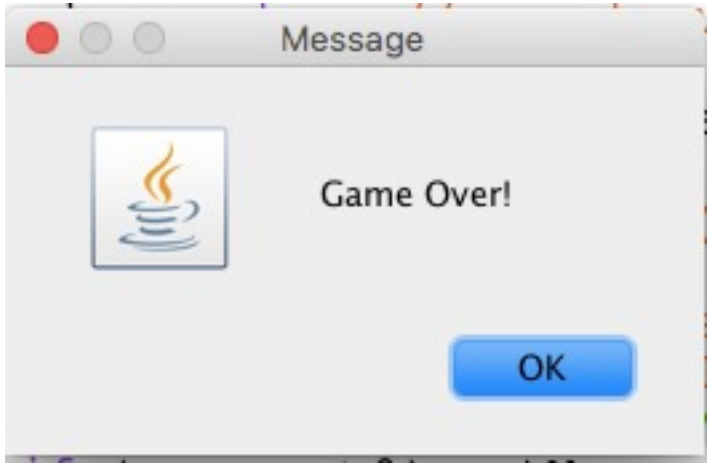
Check input | Mac version | Lab 3 | Part 2– ***extra***





Quit?





CSUN CALIFORNIA STATE UNIVERSITY NORTHRIDGE

COMP110L

Check for Valid Input

DR JEFF SOFTWARE INDIE APP DEVELOPER

Lab 3 Part 2

```
19      //start "while" loop
20    while(!done) {//continue until done
21      //popup box:  "guess the secret name:"
22        String input = JOptionPane.showInputDialog(null,"Enter Your Guess:");
23        //check for valid input
24        if (input == null) {
25           outMsg = "Canceled input";//Cancel
26           invalid = true;}
27        else if (input.equals(empty)) {
28           outMsg = "Blank input";//empty
2            invalid = true;}
3         if (invalid) {
3            if (guesses <1) {
                done = true;//or set guesses++
             outMsg += "\nYou Lost!  Good-bye";}
34           JOptionPane.showMessageDialog(null,outMsg);
35           invalid = false;//reset
36           continue;}//System.exit(0);}//or repeat?
37        //end check:  input is valid
38        //  **add code:  test if guess correct; set "win"
39        //2 cases:  win, incorrect ->NO lose
40      if (win) {
41        outMsg = "You WON!  Good-bye";
42        done = true;}//or use break
43      else outMsg = "Incorrect!";
44      //Print GUI Msg
45      JOptionPane.showMessageDialog(null,outMsg);
```

check input

bonus

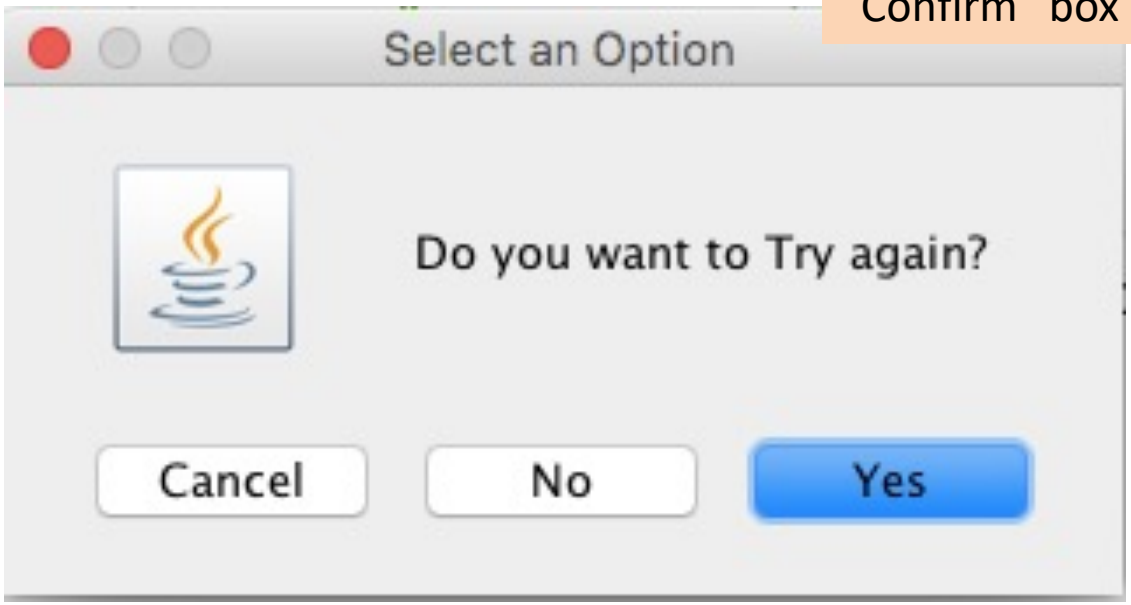
OK for part 2?

I/O: Ask

"Input" boxes

Lab 3 Part 2

"Confirm" box



CSUN
CALIFORNIA STATE UNIVERSITY NORTHRIDGE
COMP110L

Ask Continue?

DR JEFF SOFTWARE
INDIE APP DEVELOPER

Enumerative version: all cases | Lab 3 Part 2

Switch

- continuing in "WHILE" loop

```
46      //**PART 2: ask user if want to continue?
47      int cont = JOptionPane.showConfirmDialog(null,"Do you want to Try again?");
48      switch (cont) {
49        case 0://YES-- continue loop
50        break;//end switch (not loop)
51        case 1://NO-- exit loop
52           JOptionPane.showMessageDialog(null,"Game Over!  Good-bye");
53           done = true;//or use System.exit(0)
54        break;//end switch
55        default: //Cancel or Close--??
56      }//end switch
57    }//end loop
58    if($DEBUG) System.out.println("ENDing main...");
59  }//end main
60  }//end class
```

end "WHILE" loop

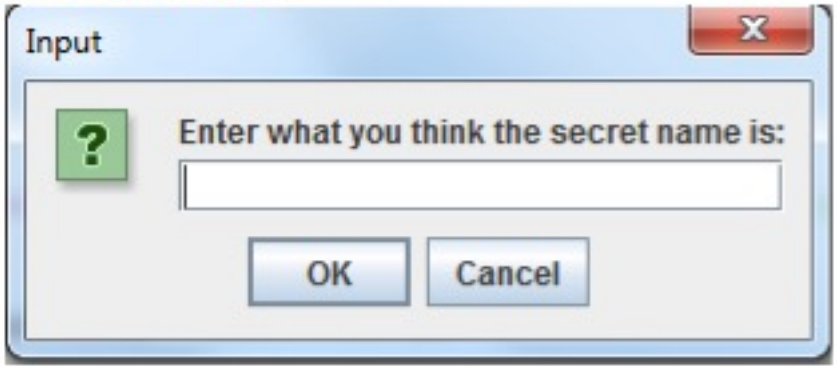
Other Examples

Win PC version | Lab 3

```
----jGRASP exec: java Secret
Input secret name:input
```

"Console" input

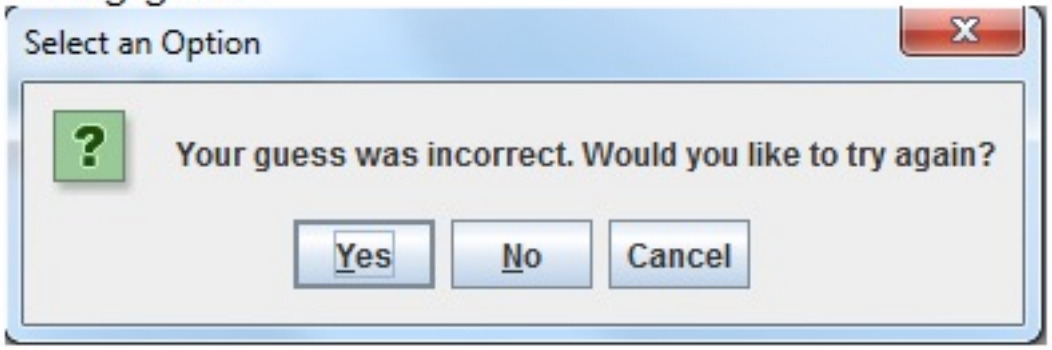
Guess window



"Input" box

Outputs

Wrong guess



"Confirm" box

Lab

Project 1

Thermostat Simulation

Projects

❖Project 1: Embedded Control

DUE AT MIDTERM

- ➢ **Thermostat** → use Temp Conversion
  - ❑ Others
    - TV remote
    - Car transmission/acceleration
    - Any other approved application

➢ while (true)

❖ Required extras
❑ USER GUIDE

❖Project 2: Simulation

DUE AT FINAL

- ➢ **Card game** → use "Shuffling"
  - **Blackjack**
  - **Poker (pick a variety)**
  - **Thermonuclear War**
  - ❑ Others
    - Weather → use Temp Conversion
    - Stock Market → ref my app (SMM)
    - US Economy (GDP, CPI, etc.)

❖ Classes

➢ game playing
- random numbers
- *monte carlo*

❖ Required extras
❑ USER GUIDE
❑ **UML**

Project Form

SYLLABUS

COMP 110/110L Intro to Algorithms and Problem Solving Fall 2016

Student: <your name> **Instructor: Dr. Jeff Drobman**

Project 1: <title>

PROJECT FORM

Description

Description

Requirements – Functions & Features

Inputs

<screen shots of all possible User Inputs>

Outputs

<screen shots of all possible Outputs>

Process (algorithms)

<screen shot of SOURCE CODE>

- UML

USER GUIDE

- User Guide

Project 1

Hennessy & Patterson

Figure 1.1.4: Embedded computer: Thermostat.

Source: zyBooks

Project 1: Thermostat

❖ Required Functions

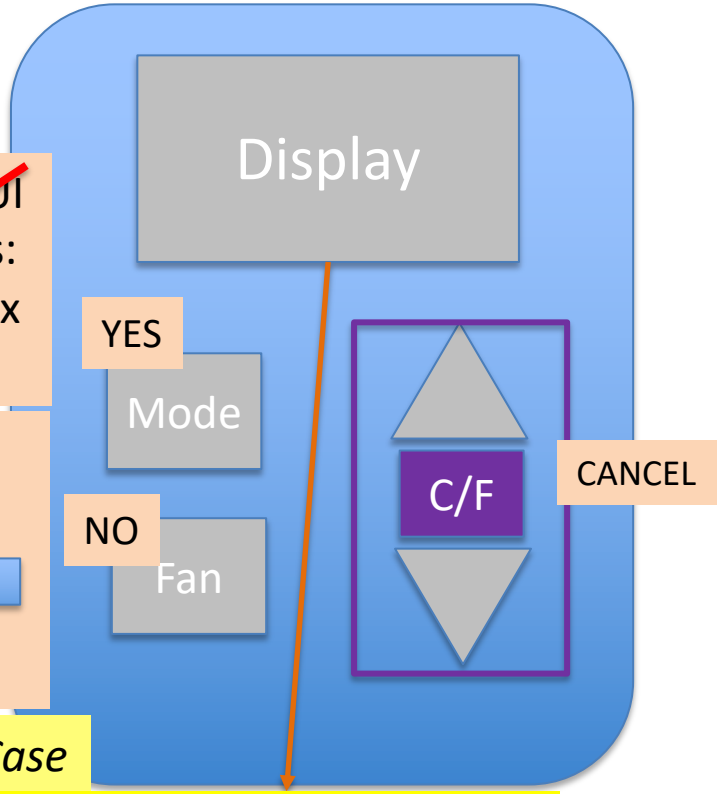
- ➢ Control buttons
  - ❑ Temperature
    - ▪ Up
    - ▪ Down
  - ❑ Mode
    - ▪ Heat
    - ▪ Cool
    - ▪ Off
    - ▪ Auto
  - ❑ Fan
    - ▪ On
    - ▪ Off
    - ▪ Auto
- ➢ Displays
  - ❑ Temperatures
    - ▪ Current
    - ▪ Set to (2: Cool, Heat)
  - ❑ Mode
  - ❑ Fan status

❖ Use Methods for C<->F

➢ Add **Time** *(extra credit)*

➢ ~~Don't use GUI~~ **Confirm** Dialogs:
1 Mode/Fan/box
2 Up/Dn/CF

➢ Use GUI **Option** Dialogs:
1 Mode
2 Fan
3 Up/Dn/C-F



➢ Use *Switch-Case*

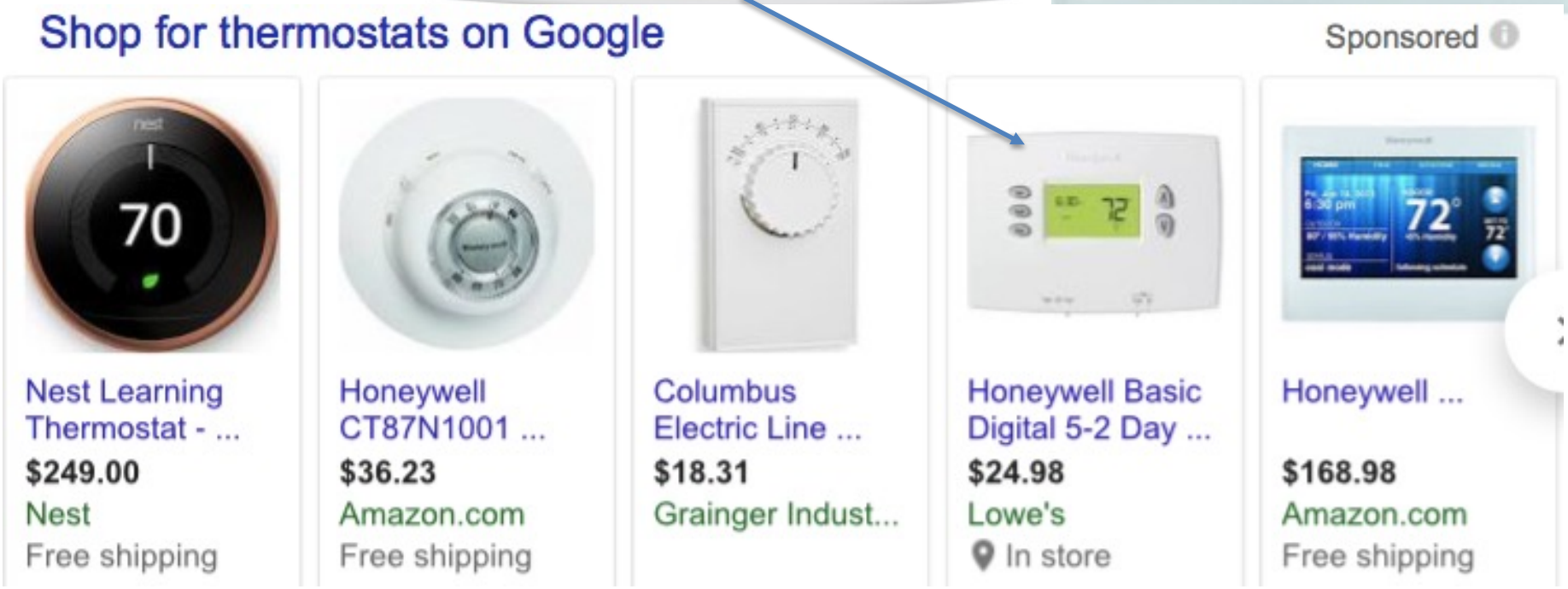
➢ Use Console – as "Display"

Temperatures:
Current: **76F**
Set to: **72F**
Mode: Cool
Fan: Auto

-OR-

Temperatures:
Current: **25C**
Set to: **22C**
Mode: Cool
Fan: Auto

Example Thermostats



Sample Code: Globals

- Global (static) variables

(Data Fields)

```
public class Thermostat {

  static String mode = "Off";
  static String fan = "Off";
  static int temper = 72;
  static int curtemp = (int)temper;
  static String ForC = "F";
```

Float/Double?

- Better

```
11 //main class
12 public class Proj1 {
13    static final boolean $DEBUG = true;
14    //initial values (at startup)
15    static double curTemp = 72;//fixed but C<>F
16    static double setTemp = 75;
17    static int mode = 0, fan = 0;
18    static char CF = 'F';//char or String?
19    //main method
```

Float/Double?

Sample Code: I/O

Confirm

```
25      int m = JOptionPane.showConfirmDialog(null, "Press
YES to change the Mode, press NO to change the Fan or CANCEL for
Temp settings.", " ", JOptionPane.YES_NO_CANCEL_OPTION);
26
27      if (m == 0) {
28          if (mode.equals("Cool")) {
29              mode = "Heat";
30          }
31          else if (mode.equals("Heat")) {
32              mode = "Off";
33          }
34          else if (mode.equals("Off")) {
35              mode = "Auto";
36          }
37          else {
38              mode = "Cool";
```

Rotate modes:
Cool →
Heat →
Auto →
Off →

- *Don't use this*
  - ❑ Use **Case-Switch**
  - ❑ Use array indexing

Sample Code

Option

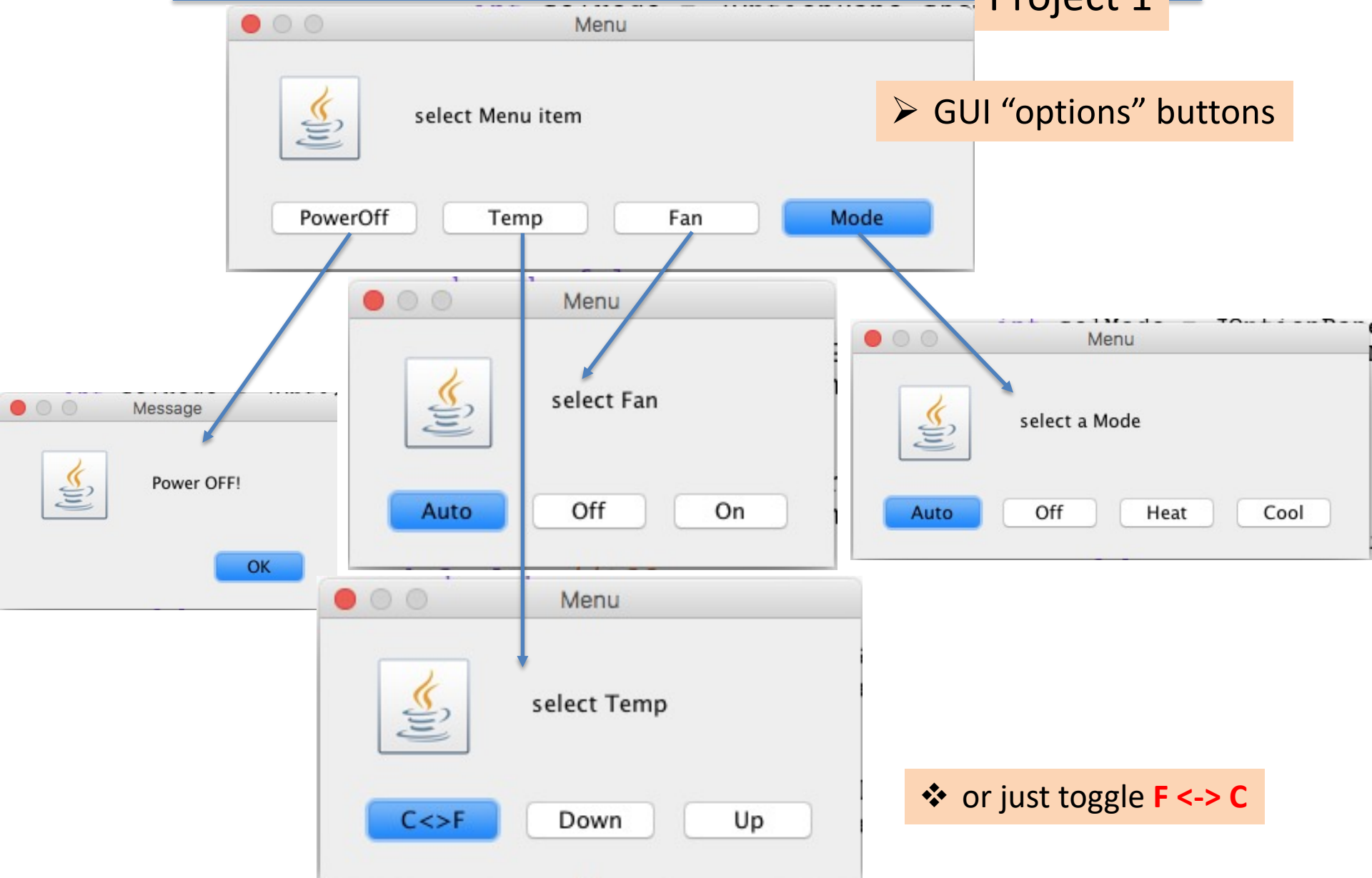
```
18      //code starts here
19      boolean run = true;
20      //create buttons
21      String[] menuMain = {"Mode", "Fan", "Temp", "PowerOff"};
22      String[] menuMode = {"Cool", "Heat", "Off", "Auto"};
23      String[] menuFan = {"On", "Off", "Auto"};
24      String[] menuTemp = {"Up", "Down", "C<>F"};
25      while(run){
26         int selMain = JOptionPane.showOptionDialog(null, "select Menu item",
27         "Menu", 0, 3, null, menuMain, menuMain[0]);
28         switch(selMain){
29            case 0: //Mode
30               int selMode = JOptionPane.showOptionDialog(null, "select a Mode",
31               "Menu", 0, 3, null, menuMode, menuMode[3]);
32               break;
33            case 1: //Fan
34               break;
35            case 2: //Temp
36               break;
37            default: //Off
38               JOptionPane.showMessageDialog(null,"Power OFF!");
39               run = false;
40         }//end switch
```

```
//create buttons
String[] menuMain = {"Mode", "Fan", "Temp", "PowerOff"};
String[] menuMode = {"Auto", "Cool", "Heat", "Off"};
String[] menuFan = {"Auto", "On", "Off"};//auto 1st=startup
String[] menuTemp = {"Up", "Down", "C<>F"};
```

Sample Menus

Project 1

- GUI "options" buttons



- or just toggle **F <-> C**

Thermostats

Figure 1 The Honeywell T-86 thermostat, better known as "The Round," is elegant and simple from both appearance and engineering perspectives; tens of millions have been installed since its introduction in 1953 and many are still in use. (Source: Cooper Hewitt Smithsonian Design Museum)

- Uses thermo-expanding coil
- Hysteresis

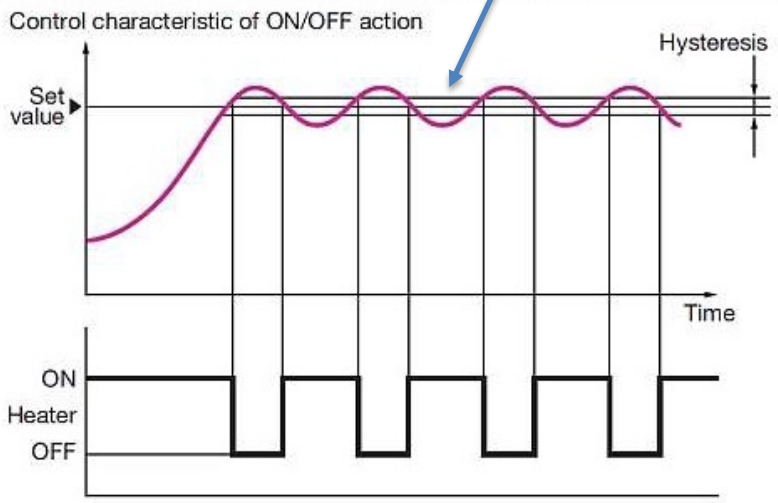


Figure 3 The standard solution is well-known: add some hysteresis around the setpoint, at the "cost" of a wider error band. (Source: Fuji Electric France S.A.S.)