THE UNIVERSITY of NORTH CAROLINA at CHAPEL HILL

COMP 110
Introduction to Programming

Thursday September 9, 2014

Jay Aikat
Fall 2014
TR 9:30 – 10:45, GS-G100

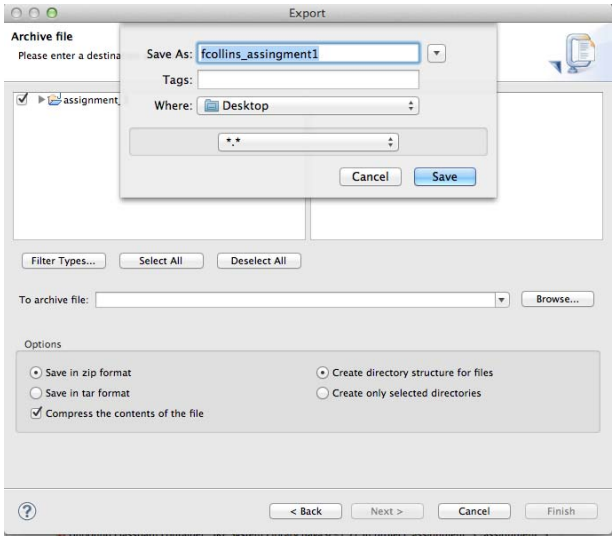
Previous Class

- What did we discuss?

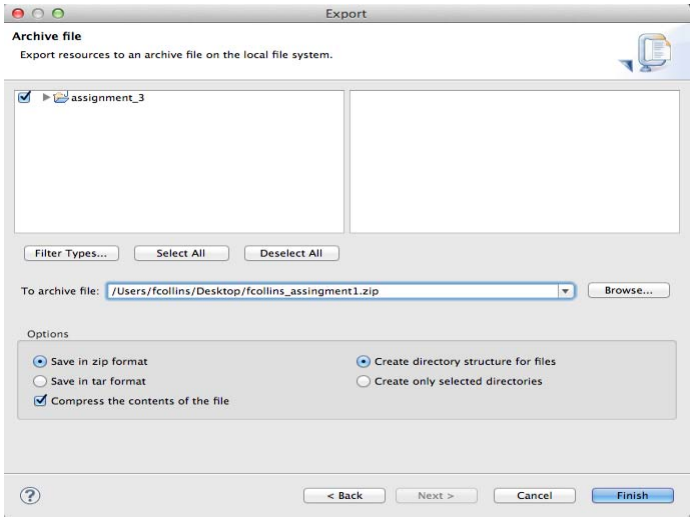
Announcements

- Assignment1 resubmission DUE today
- Lab 1 DUE Thu, 9/11
- Readings: 3.1, 3.2
- **Assignment 2 assigned today; DUE Tue, Sep 16 @ 11:55 PM**

Saving on a Mac



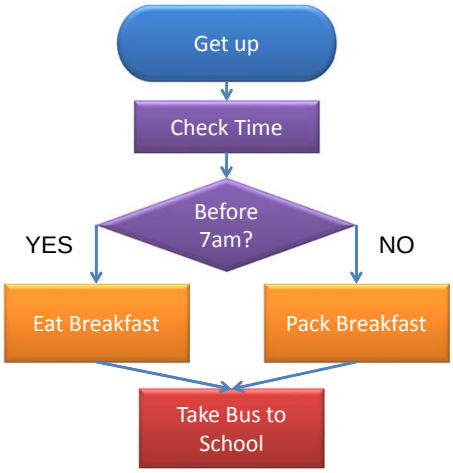
Saving on a Mac



Flow of Control

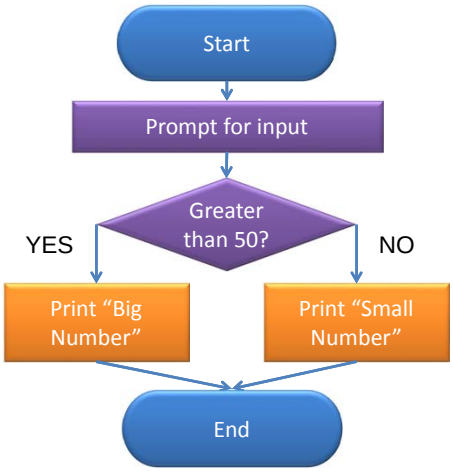
- *Flow of control* is the order in which a program performs actions.
  - Up to this point, the order has been sequential.
- A *branching statement* chooses between two or more possible actions.
- A *loop statement* repeats an action until a stopping condition occurs.

Flow Chart




```
Student.getUp();
if (time < 7) {
  Student.eatBreakfast();
}
else { // time >= 7
  Student.packBreakfast();
}
Student.takeBus();
```


Java Example




```
import java.util.*;

public class FlowChart {
  public static void main(String[] args) {
    System.out.println("Give me an integer:");
    Scanner keyboard = new Scanner(System.in);
    int inputInt = keyboard.nextInt();
    if (inputInt > 50)
    {
      System.out.println("Big number");
    }
    else
    {
      System.out.println("Small number");
    }
  }
}
```

What if your input is exactly 50?

If and Else

– Take two minutes and write down an example pseudocode of one If and Else Statement

– For example:

```
if (avgQuizGrade >= 90) {
        Topic.startNext();
}
else {
        Topic.redoLast();
}
```


Java Comparison Operators

FIGURE 3.4 Java comparison operators

Math Notation	Name	Java Notation	Java Examples
=	Equal to	==	`balance == 0` `answer == 'y'`
≠	Not equal to	!=	`income != tax` `answer != 'y'`
>	Greater than	>	`expenses > income`
≥	Greater than or equal to	>=	`points >= 60`
<	Less than	<	`pressure < max`
≤	Less than or equal to	<=	`expenses <= income`

Expressions

- Expression?
  - An **expression** can be a variable, a value, or a combination made up of variables, values and operators
  - An expression **has a value**
  - **Arithmetic expression**: a combination of numbers with a number value
    - *10, taxRate/100, (cost + tax) * discount*
  - **String expression**: a combination of Strings with a String value
    - *"Hello", "The total cost is " + totalCost*

Boolean Expressions

- A combination of values and variables by comparison operators. Its value can only be ***true*** or ***false***
- Example expressions
  - *5 == 3; // false*
  - *variable <= 6; // depending on the value of variable*
    - What if variable is 5? What if variable is 6?
  - *myInt != temp; // depending on both values*
    - What if myInt is 0 and temp is 2?

Syntax for IF statement

- Syntax rule for if statement:
  - *if (boolean expression)*
    { statements; }

&&: and

- What if you need multiple expressions to be true?
- Syntax rule:
  - *(expression) && (expression) && …*
    - Expressions go in ( )
  - (Time < 7) && (I've prepared breakfast)
- Will only be true if **ALL** statements are true

||: or

- What if you need ONE expression to be true out of many expressions
- Syntax rule:
  - ***(expression)*** **||** ***(expression)*** **|| …**
    - Again, expressions go in ( )
  - (I've had breakfast) || (Time > 7)
- Will be true if **ONE** expression is true

!: not

- Syntax rule:
  - ***!(expression)***
    - Again, expressions go in ( )
  - !(I've had breakfast)
- Will be **true** if the expression is **false**
- **! is not recommended**
  - **You will get confused. Try to write expressions straightforward**
    - Use (cost != 3) instead of !(cost == 3)
    - Use (time <= 7) instead of !(time > 7)

Logical Operators

FIGURE 3.7 The Effect of the Boolean Operators && (*and*), || (*or*), and ! (*not*) on Boolean Values

Value of *A*	Value of *B*	Value of *A* && *B*	Value of *A* || *B*	Value of ! (*A*)
true	true	true	true	false
true	false	false	true	false
false	true	false	true	true
false	false	false	false	true

Comparison vs. Logical Operators

- Comparison operators connect values or variables
  - After connection, it's a boolean expression
  - *a > b*
  - *c == d*
- Logical operators connect boolean expressions
  - *(a > b) && (c == d)*

More Complex Boolean Expressions

- Combination of && and ||
  - *(((3 < 7)||(2==5)) && ((4!=2) && (1 <= 1)))*
  - *(((true)||(false)) && ((true) && (true))*
  - *(true) && (true)*
  - *true*
- ***if*** *(((I'm at Subway) && (You're at Subway)) ||*
  ((I'm at Starbucks) && (You're at Starbucks))

```
{
    I will see you;
}
```


Boolean Variable

- A boolean variable saves a boolean value

```
boolean systemsAreOK =
  ((temperature <= 100) && (thrust >= 12000) && (cabinPressure > 30));
// You can use "=" to assign a boolean value to a boolean variable
if (systemsAreOK){
// It's the same as if (systemsAreOK == true)
   System.out.println("Initiate launch sequence.");
   }
else{
   System.out.println("Abort launch sequence.");
   }
```

Assignment vs. Equal To

- *if (n1 = n2)*
  - **Error!!!!** It's an assignment statement!
- *if (n1 == n2)*
  - Correct. It's a boolean expression now.

Next class (Thu, Sep 11)

- Prof. Gary Bishop – guest lecture!