1.00 Tutorial 1

Introduction to 1.00

Outline

- Introductions
- Administrative Stuff
- PS 0
- Java Basics
- Eclipse practice
- PS1 discussion

Administrative stuff (1)

Top five reasons why you should attend tutorials:

- You get the opportunity for individualized instruction and interactive discussion of questions and concepts.
- You get to write some code in a cozy atmosphere
- You get to speak and ask questions (don't be shy!)
- You get to review some key points from lecture, to discuss problem sets, and to practice for quizzes and final exam
- Mandatory (if you want full credit– 6%)
  - Grade based on participation

Administrative stuff (2)

Getting help

THE GOLDEN RULES:

1- Start **EARLY**, we mean **REALLY early**.

2- Do not start a problem set without making sure that you fully understand the lecture concepts. Homework is just an application of key lecture ideas.

3- The debugger is there for a reason. Use it! It can be really fun, and would spare you hours of hassle.

- Feel free to come with general questions about Java, software, lectures, tutorials or homework. However, we will **NOT**:

  - Dedicate the entire afternoon to you. There are 200 people taking the class, sorry!

  - Read the problem set for you. If you haven't attempted it before, you will be asked to leave (we look sweet, but we can be rude at times)

  - Write any lines of code for you. This is what tutorials and active learning sessions are for.

  - Answer questions of the type: "I have the following 200 lines of code. Why isn't my program running correctly?". We are neither debuggers nor prophets.

Administrative stuff (3)

Laptop Problems/etc

- Use a power cable when you can
- Always back up your work, either using CDR's or USB Flash drives
- Please remember to put a comment in your code with your name, email id, TA & Section Name

PS 0

- Academic Honesty
  - It’s less work to do the problem sets than to cheat without getting caught.
- Introduction to Eclipse
  - Start Eclipse now!
- Submitting homework and printing
  - Ask us before/after lecture or go to office hours this week if you’re having problems

Java Basics (1)

- *Declaring variables*
  - initial value required? what about type?

  -- a variable is simply a memory location that stores a value.
- *Assigning a value vs. testing a value*
  - compiler will catch this, but know the difference (= v/s ==)
- *Declaring floats and longs (F / L)*
- *Representing booleans*
  - use true/false, not 0/1
- *Naming conventions*
  - Java is case-sensitive!
  - classes and filenames: always Capitalize
  - variable names: int runningSpeed=55;
  - must begin with letter, underscore, or $
  - final variables: final double PI = 3.1416;

Java Basics (2) : fundamental types

type	bits
boolean	1
byte	8
char	16
short	16
int	32
long	64
float	32
double	64

Java Basics (3): examples

```
int studentCount = 91;
char firstLetter = 'a';
float weight = 180.6F;
double area = Math.PI * 5.0 * 5.0;
boolean enjoy100 = true;
boolean xGreaterThanY = (x > y);
long theNumberOne = 1L;
double googol = 1E100; //10^100
```

- Make sure you don't use Java keywords (`do`, `while`, `import`…) as variable names!

Java Basics (4): type conversion

- Implicit (automatic) conversion is called *promotion:*

int promoted to double

```
double x = 3.1415 + 2;
```

- Explicit conversion is called *casting*

```
int myAge =
     (int)(Math.PI*10.0+4.0);
```

expression cast to int

this expression is a `double` (very accurate)

Java Basics (5): promotion

increasing capacity ⟶

Data Type	Allowed Promotions
`double`	*None*
`float`	`double`
`long`	`float,double`
`int`	`long,float,double`
`char`	`int,long,float,double`
`short`	`int, long,float,double`
`byte`	`short,int,long,float,double`

Eclipse practice (1)
Setup

- On your hard drive, create a folder that will hold all your Java files throughout the semester.
- Start Eclipse by double-clicking on its icon
- Create a new project named *Tutorial1* or *Tut1*. Project names have nothing to do with actual folder names.
- Under this project, create a new class called *GettingInput.* It should have a `main()` method.

Eclipse practice (2)

Variables, control structures, debugger

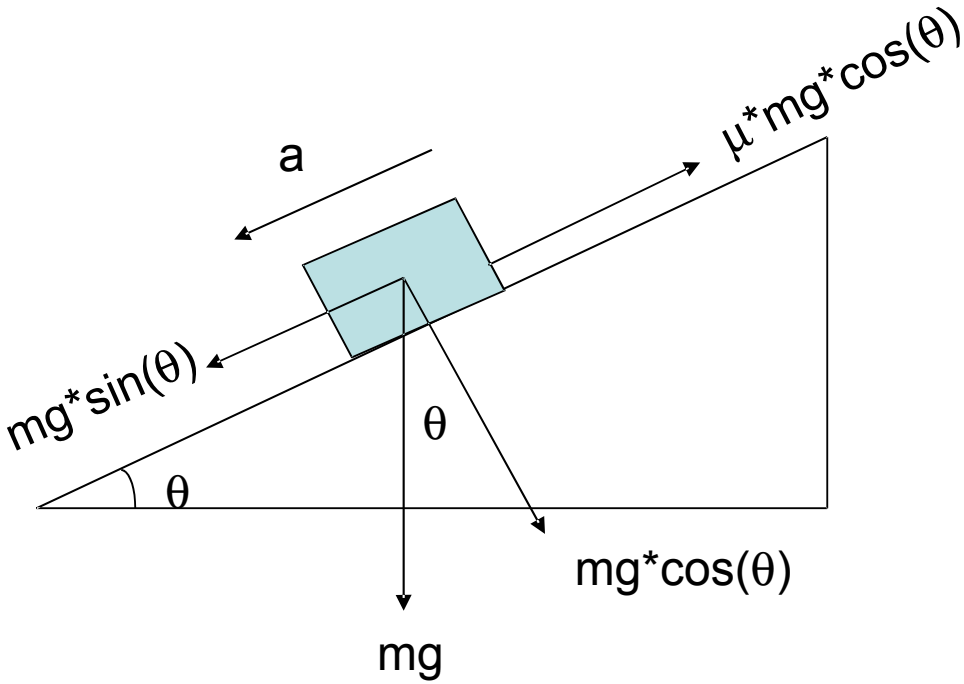
- Declare a `long` and set it equal to 1000
- Declare a `float` and set it equal to 2.33
- Write a `for` loop that starts from 1 and increments the counter `i` by 2 in each iteration. Print the value of the counter at every iteration.
- Use the debugger to trace the values of all variables in your program. Don’t forget to use a breakpoint.
- Replace the `for` loop by an equivalent `while` loop

Eclipse practice (3)

JOptionPane

- Ask the user for temperature in Fahrenheit using a `JOptionPane`
- `import javax.swing.*;`
- Store the input in a variable `tempF`
- Convert it to Celsius and print it out using `System.out.println()`
- `tempC = (tempF-32)*5/9`

Problem Set 1



Force equilibrium along the plane

$ma = mg*\sin(\theta)-\mu*m*g\cos(\theta)$

Problem Set Practice

- Ask the user to enter two masses m1 and m2
- If m1 is greater than m2 then swap their values

PS1 Sample Output

Command Prompt

```
C:\Documents and Settings\LLP\Desktop\homework\HW1S05>java SlideLength
Boxes in wrong order; corrected
Results
Slide elevation(m): 10.0
Max velocity(m/sec): 2.0
Box 1 mass(kg): 4.0
Box 2 mass(kg): 2.0
Box 3 mass(kg): 1.0
Acceleration(m/sec^2): 0.29849630656569165
Tension 1(kg-m/sec^2): 0.7919139190438107
Tension 2(kg-m/sec^2): 0.2639713063479368
Theta (radians):0.3400000000000014
Theta (degrees): 19.480565034448
Slide length(m): 29.98616808766173
Time to reach bottom(sec): 14.17444302180697
Velocity at bottom(m/sec): 4.231018889635222

C:\Documents and Settings\LLP\Desktop\homework\HW1S05>_
```