Today

• Review of Java Programming Language
• Programming environments
• Debugging
• Language constructs
• Input and output
• GUI

• More on Logistics
Java

• What are some of the advantages of Java compared to other languages?
Java

• What are some of the advantages of Java compared to other languages?

• What are some of the disadvantages of Java?
Java is “Platform-Independent”

- C source code
- Unix Compiler
- MacOS Compiler
- Windows Compiler
- Binary code
  - Unix
  - MacOS
  - Windows

- Java source code
- Java Compiler
- Byte Code
- Java Virtual Machine (Interpreter)
What Is A “Class”?

Definition of a **container** holding data and methods (functions)

```java
class <classname> { 

<variable declarations> 
    ... 
<method declarations> 
    ... 
<optional main method> 

}
```
What Is A “Class”? 

Example: 
Define a class Apple.

What’s the difference between the class definition and its instance?

What’s the purpose of the main function?
Simple Example

- **HelloWorld.java**

```java
class HelloWorld {
    public static void main(String args[]) {
        System.out.println("Hello, World!");
    }
}
```

File name MUST match class name.
Compile/Run

• Command to compile a Java class:
  - javac HelloWorld.java

• Command to execute the main method of a Java class:
  - java HelloWorld
Compile/Run

• Using Integrated Development Environment makes your life much easier!

• We recommend Eclipse or NetBeans
  – Cross platform
  – Easy to program, debug, linking to jar files, getting document help...
Statistics

• How many of you will work on your assignments in:
  – Windows
  – Mac
  – Linux
“Primitive” Data Types

• Integral (not a class!)
  - int i=100;

• Decimal (not a class!)
  - float height;
  - double weight;

• Logical (not a class!)
  - boolean isover;

• Character (not a class!)
  - char answer = ‘y’;
Wrapper Classes

• Each primitive type has a “wrapper class”.
  - Integer, Double, Character, Boolean ...
  - Contain added functionality.

• Example: String Class
  - String message;
  - String className = “ece242”;
  - className.contains(“ece”)
Expressions

• Add/Subtract
  - \( x = y + 3 \);
  - \( m = n - 4 \);

• Multiply/Divide
  - \( x = y / 3 \);
  - \( m = 2 \times 6 \);

• Boolean
  - \( \text{done} = (x==y) \&\& ((!z>10) \| (z>=1)) \);
Conversion between types

• Is this correct?
  ```
  double a = 5;
  ```

• Is this correct?
  ```
  int a = 5.0;
  ```

• What’s the value of a below?
  ```
  double a = 3/5;
  ```
Conversion between types

• Is this correct?
  \[\text{double } a = 5; \quad \rightarrow \quad \text{correct}\]

• Is this correct?
  \[\text{int } a = 5.0; \quad \rightarrow \quad \text{incorrect}\]

• What’s the value of \(a\) below?
  \[\text{double } a = \frac{3}{5}; \quad \rightarrow \quad a = 0.0\]
Conversion between types

Low precision -> high precision
no problem, automatic conversion

High precision -> low precision
needs explicit conversion
float a = 5.5;
int b = (int)a;
Conversion between types

• Examples:

```java
int x = 5;
float y, z;
y = x/2;
z = (float)x/2;
```

• Results for y and z are different
Conversion between types

• Examples:
  
  ```
  int x = 5;
  float y, z;
  y = x/2;
  z = (float)x/2;
  ```

• Results for y and z are different
  
  • \( y = 2; \)
  • \( z = 2.5; \)
Flow Control Statements

• *if-else* statements
• *for* loop
• *while* loop
• *do-while* loop
if-else branching statements

if (<condition>)
{
   <statement 1>;
   ...
}
else
{
   <statement 2>;
   ...
}
if-else statements

• Given a positive integer n, is it odd or even?

if (n%2==0)
    System.out.println(n+" is even");
else
    System.out.println(n+" is odd");
if-else statements

• Another example:
  – Grading policy based on one exam.
    
    85\leq \text{your score}\leq 100, \text{you will get grade A,}
    
    75\leq \text{your score} < 85, \text{you will get grade B;}
    
    65\leq \text{your score} < 75, \text{you will get grade C;}
    
    60\leq \text{your score} < 65, \text{you will get grade D;}
    
    Otherwise, you will get grade E;
if (yourscore>=85 && yourscore<=100)  
    yourGrade = 'A';  
else if (yourscore>=75 && yourscore<85)  
    yourGrade = 'B';  
else if (yourscore>=65 && yourscore<75)  
    yourGrade = 'C';  
else if (yourscore>=60 && yourscore<65)  
    yourGrade = 'D';  
else  
    yourGrade = 'E';
if-else statements

if (yourscore>=85)
    yourGrade = 'A';
else if (yourscore>=75)
    yourGrade = 'B';
else if (yourscore>=65)
    yourGrade = 'C';
else if (yourscore>=60)
    yourGrade = 'D';
else
    yourGrade = 'E';
if-else statements

if (choice == 1)
    order = "scrambled egg";
else if (choice == 2)
    order = "blueberry pancake";
else if (choice == 3)
    order = "italian sausage";
else
    order = "bacon";
switch-case statements

switch(order) {
    case 1:
        order = "scrambled egg";
    case 2:
        order = "blueberry pancake";
    case 3:
        order = "italian sausage";
    default:
        order = "bacon";
}

What’s the problem here?
switch-case statements

switch (order) {
    case 1:
        order = "scrambled egg";
        break;
    case 2:
        order = "blueberry pancake";
        break;
    case 3:
        order = "italian sausage";
        break;
    default:
        order = "bacon";
}
Looping Constructs

• *for* loop
  – usually uses a counter for control

• *while* and *do-while* loops
  – usually uses a Boolean statement for control
for loop

for( <initialize counter>; <condition>; <update counter>)
{
    <statement 1>
    <statement 2>
    ... ...
}

for loop

Initialization

Condition

for loop body
Update expression
A simple example

Print \( n \) asterisks

```java
for (i=0 ; i<n ; i++)
{
    System.out.println( "*" );
}
```
Display Rectangle

• Display a 3x4 rectangle formed by *

****
****
****
****
Nested for loop

- ForLoopDisplayRectangle.java

```java
for (i=0; i<3; i++)
{
    for(j=0; j<4; j++)
        System.out.print("*");
    System.out.println();
}
```
Multiplication Table

- Print a Multiplication Table

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>4</td>
<td>6</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>6</td>
<td>9</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>8</td>
<td>12</td>
</tr>
</tbody>
</table>
Multiplication Table

```java
int ROWS = 4;
int COLS = 3
int row, col;
for (row=1 ; row<=ROWS;  row++)
{
    for (col=1 ; col<=COLS;  col++)
    {
        System.out.print(" "+row*col+" ");
    }
}
```
Two Loop Constructs

• **while** loop
  – Check loop condition
  – Then loop body

• **do-while** loop
  – Runs loop body first
  – Then checks condition
  – runs body at least once
while statement

while (<condition>)
{
    <statement1>;
    <statement2>;
}

Condition

yes

no

Statement 1
Statement 2
do-{ 
  <statement1>; 
  <statement2>; 
} while (<condition>);
Compounding Interest

• Suppose you have $1,000 in a savings account that earns 4% interest per year.

• How many years until you double your money?
Compounding Interest

double my_money = 1000.0;
int n = 0;
while (my_money < 2000.0) {
    my_money = my_money * 1.04;
    n = n + 1;
}
System.out.println("My money will double in "+ n + " years");
Arrays

An array is a contiguous piece of storage for elements of compatible types; it’s just a reference:

```java
int[] numbers = new int[50];
for (int i = 0; i < numbers.length; i++) {
    numbers[i] = i;
}
```

numbers.length == 50
Input/Output

• InputOutput.java
  – Input your name, print out your name
  – Input your age, print out our age
  – Input your height, print out your height
  – Using Scanner.java
import java.util.*;

class InputOutput {
    public static void main(String args[]) {
        Scanner S = new Scanner(System.in);
        String name;
        System.out.print("enter your name: ");
        name = S.next();
        System.out.println("Your name is: "+ name);
    }
}
InputOutput.java

```java
int age;
System.out.print("enter your age: ");
age = S.nextInt();
System.out.println("Your age is: " + age );

float height;
System.out.print("enter your height(feet): ");
height = S.nextFloat();
System.out.println("Your height is: " + height + " feet");
}
```
Logistics

• TAs
• Office hours
• Submission policy
• EDLAB
• Google Calendar
Coming up

• Assignment 1 is due by Thursday next week
  – Description will be finalized by tonight
• Discussion section Wednesday next week
  – Quizzes count into your grade