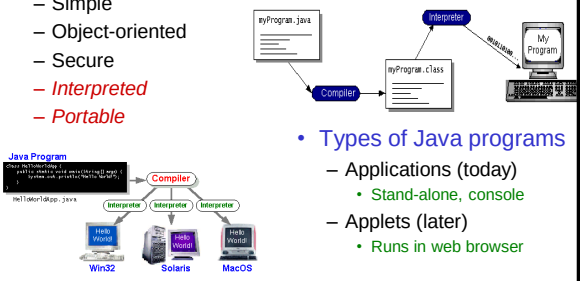
CS 325 - Class 14

- Today
  - Java basics
  - Building simple Java programs
  - Command-line arguments in Java
  - Using JDK and Eclipse
- Announcements
  - Continue working on Project 3

Introduction to Java

- Basic Principles
  - Simple
  - Object-oriented
  - Secure
  - *Interpreted*
  - *Portable*



- Compilation process

- Types of Java programs
  - Applications (today)
    - Stand-alone, console
  - Applets (later)
    - Runs in web browser

Running a simple program

- On **bama.ua.edu**
  - Create a file that contains the code shown at the bottom
  - Name your file **RollTide.java**
- To compile it, type
  - **javac RollTide.java**
  - Creates a RollTide.class file
- To interpret it, type
  - **java RollTide**

```
public class RollTide {
   public static void main  (String [ ] args)  {
        System.out.println("Roll Tide");
   }
}
```


Class Exercise

- Repeat the process described on the previous page, only this time compile it in a Windows environment
  - Use notepad to create the file
  - Invoke a console window (command prompt)
    - Compile using **javac**
    - Interpret with **java**

Example: Java versus C++

Java

```
public class Sum10 {
    public static void main (String[ ] args) {
       int sum = 0;
       for (int iter = 1; iter <= 10; ++iter) { sum += iter; }
       System.out.println("Sum = " + sum);     // also have System.out.print( … )
    }
}
```

C++

```
#include <iostream>
using namespace std;
int main( int argc, char *argv [ ] ) {
    int sum = 0;
    for (int iter = 1; iter <=10; ++iter) { sum += iter; }
    cout << "Sum = " << sum << endl;
}
```


Class Exercises

- Write a Java program that prints out the numbers 5 10 15 20 … 95 100
- Write a Java program that computes and prints the sum of the first 100 positive integers

Java basics (slide 1 of 4)

Data Types

- boolean (bool in C++)
- char (2 bytes, 16 bits)
- byte (8 bits)
- short (16-bit int)
- int (32-bit int, C++)
- long (64-bit int)
  9,223,372,036,854,775,808
- float (4 bytes)
- double (8 bytes)
- String

Operators

- ++ and – –
- !
- * / %
- + –
- << >> (bit shift)
- < > <= >=
- == !=
- &&
- ||
- = *= /= %= += –=

Java basics (slide 2 of 4)

- Selection Statements

```
if (conditional) {
   stmt(s);
}
else {
   stmt(s);
}
```

  - else optional
  - { and } only needed if multiple statements

- Switch statement

```
switch (variable) {
   case value1: stmt(s); break;
   case value2: stmt(s); break;
   default: stmt(s); break;
}
```

- Iteration Statements

```
for (initial condition;
        test for continuation;
        increment) {
              stmt(s);
}

while (condition) {
        stmt(s);
}

do {
        stmt(s);
} while (condition);
```


Java basics (slide 3 of 4)

- C++, storage allocated during declaration
  - **int data[100];**
- Java, declaration is different from storage allocation
  - **int [ ] data;**
  - **data = new int[100];**
- Can combine declaration and allocation

  int [ ] data = new int[100];

- Subscripts start at zero
- Array has a built-in field called ***length***
  - **char[ ] ch = new char[5];**
  - ***ch.length***

  for (i=0; i<ch.length; i++) {
  …
  }

- Initialization

  int [ ] data = {v_1, v_2, … v_n };

Java basics (slide 4 of 4)

- Java Strings
  - http://java.sun.com/j2se/1.5.0/docs/api/java/lang/String.html
- Declaring and Constructing String
  - String fred;
  - String fred = "xxxx";
  - String fred = null;

- Basic String Methods
  - char charAt(int)
  - boolean equals(object)
  - int indexOf(char)
  - int indexOf(String)
  - String substring(int)
  - String substring(int, int)

String fred;
fred = new String("UofA");
xxx = fred.***method***(***args***)

Class Exercises

- Write two Java programs to print out the two patterns shown below using nested loops:

```
X          XXXXXXXXXX
XX          XXXXXXX
XXX          XXXXX
XXXX          XXX
XXXXX          X
```

- Write a Java program that stores the numbers 1^2, 2^2, …, to 50^2 in an array, and then prints out the array in reverse order (50^2 down to 1^2)
- Write a Java program that prints the first ten Fibonacci numbers. Recall that: $F_0 = F_1 = 1$, and $F_k = F_{k-2} + F_{k-1}$

Random Numbers in Java

- We have to include some additional libraries
  - Similar to #include <…>
- Declare a new random number generator "object"
- This object can then generate random numbers for our program
- The random numbers generated are both positive and negative

```
import java.util.Random;

public class foo {
  public static void main( String[ ] args ) {
    int x;
    Random generator = new Random( );
    for ( ... ) {
      ...
      x = generator.nextInt( );
      ...
    }
  }
}
```

Class Exercises

- Generate 1,000,000 random numbers. Print out the counts of how many positive and negative numbers were generated.
- Generate 1,000,000 random numbers in the range of **-99** to **99**. Did you see more positive or negative numbers?
- Generate random numbers in the range of **0..100** until you generate a duplicate number. How many numbers were generated before you duplicated a number?
- Generate 1,000,000 random numbers in the range of **0..100**. Which number occurred most frequently?

Command Line Args in Java

```
public class CmdArgs {
  public static void main ( String [ ] args ) {
    System.out.println (args.length);
    for (int a=0; a<args.length; a++)
      System.out.println (a + " " + args[a]);
  }
}
```

- Don't need to know the number of arguments
  - String contains a built-in "length" field
  - All arguments are still strings
- This program prints out the number of each argument in addition to the argument itself

Class Exercise

- Implement the previous Java command-line argument program
- Compile and run the program

  javac *CmdArgs.cpp*

  java *CmdArgs* Homer Marge Bart Lisa Maggie

Class Exercise

- Implement and run this Java program

```
public class CmdArgs2 {
  public static void main ( String [ ] args ) {
    int sum = 0;
    for (int k=0; k<args.length; k++)
      sum += Integer.valueOf(args[k]).intValue( );
    System.out.println("Sum = " + sum);
  }
}
```

- Integer.valueOf( ) converts a String to an Integer
- intValue( ) converts an Integer to an int
- Integer denotes a class, but int is a primitive type

Installing Java on a PC

- Java Development Kit
  - http://java.sun.com/javase/downloads/index.jsp
  - Download JDK 6, most recent update
- Documentation
  - Same web page
  - Java SE 6 Documentation
  - Also access the Java reference guides and tutorials from this link

1. Install J2SE
2. Install documentation under docs directory in C:\Program Files\Java
3. Install tutorial under tutorial directory, also in C:\Program Files\Java
4. Set path appropriately as explained in http://java.sun.com/javase/6/webnotes/install/jdk/install-windows.html

Optional: Eclipse IDE for Java

- Similar to Visual Studio for C++ programs
- Software is installed in some classrooms and labs
  - Can also be downloaded for home use at http://www.eclipse.org/downloads/
- Start Eclipse
- Instructions below are for Version 3.3.0
  - Other versions might vary slightly
- Create a new Java project
  - File => New => Java Project
  - Enter a Project name
  - Select "Create separate folders for sources and class files"
  - Press "Finish" button

Eclipse IDE (cont.)

- Add a Java program to your project
  - Window => Show view => Package Explorer
  - Left click on “+” beside your current project’s name
  - Right click on “src”, then select New => Class
  - Enter the Name of your class/file
  - Press “Finish” button
  - Type a Java program in the file window
  - File => Save (the file now compiles automatically)

Eclipse IDE (cont.)

- Run your program
  - Window => Show view => Navigator
  - Run => Run As => Java Application
    - If asked to save resources, press “OK”
  - Output appears in Console
- If you want to pass arguments to your program
  - Run => Open Run Dialog…
  - Left click on your application in the left panel
    - If your application is not shown, right click on Java Application, then select New
  - Left click on Arguments tab
  - Enter program arguments separated by spaces
  - Press “Run”

Class Exercises

- Repeat any of today’s previous Java programs
  - This time use the Eclipse IDE to run your programs