CS3500: Object-Oriented Design
Fall 2013

Class 2
9.10.2013
Java Classes

Class

Method

Statement

Statement

Statement

Field

Field

Method

Statement

Statement

Statement

Method

Statement

Statement

Statement
public class <Name> {

    <field>
    <field>
    <method>
    <method>

    ...

    <method>

}
Class Declaration
[Sestoft, p.20]

class-modifiers class C extends-clause implements-clause {
  field-declarations
  constructor-declarations
  method-declarations
  class-declarations
  interface-declarations
  enum-type-declarations
  initializer-blocks
}
<modifiers> class <Name>
    extends <...> implements <...>

public class Book
Class Declaration

[Sebstoft, p.26]

method-modifiers return-type m(formal-list) throws-clause method-body {

}

}
Interface Declaration

[Sestoft, p.60]

interface-modifiers interface I extends-clause {
  field-descriptions
  method-descriptions
  class-declarations
  interface-declarations
}

Northeastern University
College of Computer and Information Science
Method Header

<modifiers> <return-type> <method-name>(<parameters>)

public static void main(String [] args)
Field Declaration

[Sestoft, p.24]

field-modifiers type fieldname1, fieldname2, ...; field-modifiers type field name = initializer, ... ;
Write-Compile-Execute

• Write
  source code: instructions in a program – human readable (.java)

• Compile
  – translates program from one language to another
  – byte code: intermediate language that can be run by many computers (.class)

• Execute
  Uses the Java Virtual Machine (JVM) to run the program
Write-Compile-Execute

- Write
  - HelloWorld.java
  - failure
- Compile
  - success
- Execute
  - HelloWorld.class
Hello World!!

/**
 * This is an example class that
 * illustrates printing a message to the
 * screen
 * @author Jessica Young Schmidt jschmidt
 */
public class Hello {
    /**
     * main method which will print message
     * @param args command line arguments
     */
    public static void main(String[] args) {
        System.out.println("Hello World!");
    }
}

• Define a class named Hello
• Define a method called main
  Main method required to start a Java program
• Defined statements of what the program should do
  In this case, print a string “Hello World!”
Static Methods
Mutation
equals and hashCode
StackInt Signature

empty:                    -->  StackInt
push:     StackInt x int  -->  StackInt
isEmpty:  StackInt        -->  boolean
top:      StackInt        -->  int
pop:      StackInt        -->  StackInt
size:     StackInt        -->  int
StackInt Algebraic Specifications

isEmpty (empty()) = true
isEmpty (push (s, n)) = false

top (push (s, n)) = n

pop (push (s, n)) = s

size (empty()) = 0
size (push (s, n)) = 1 + size (s)