Final Review

Forest Agostinelli
University of South Carolina

Final Exam Contents

- Lectures 1-14 are fair game for the final exam
    - Will not need to know details of breadth-first search or eight puzzle
- Concepts from lecture 15 will not be on the main exam, but may be asked for extra credit
- Tips for studying
    - Review all lecture slides, especially all "quick quiz" sections
    - Re-do all labs and homeworks

Data Types

Data Type	Size	Description
byte	1 byte	Stores whole numbers from -128 to 127
short	2 bytes	Stores whole numbers from -32,768 to 32,767
int	**4 bytes**	**Stores whole numbers from -2,147,483,648 to 2,147,483,647**
long	8 bytes	Stores whole numbers from -9,223,372,036,854,775,808 to 9,223,372,036,854,775,807
float	4 bytes	Stores fractional numbers. Sufficient for storing 6 to 7 decimal digits
double	**8 bytes**	**Stores fractional numbers. Sufficient for storing 15 decimal digits**
boolean	**1 bit**	**Stores true or false values**
char	**2 bytes**	**Stores a single character/letter or ASCII values**

- For a byte, how is 0 represented in binary?
- How is 1 represented in binary?
- How is -1 represented in binary?
- How is -128 represented in binary?

Data Types

Data Type	Size	Description
byte	1 byte	Stores whole numbers from -128 to 127
short	2 bytes	Stores whole numbers from -32,768 to 32,767
int	**4 bytes**	**Stores whole numbers from -2,147,483,648 to 2,147,483,647**
long	8 bytes	Stores whole numbers from -9,223,372,036,854,775,808 to 9,223,372,036,854,775,807
float	4 bytes	Stores fractional numbers. Sufficient for storing 6 to 7 decimal digits
double	**8 bytes**	**Stores fractional numbers. Sufficient for storing 15 decimal digits**
boolean	**1 bit**	**Stores true or false values**
char	**2 bytes**	**Stores a single character/letter or ASCII values**

- What value of “val” will be print to the screen

```
byte val = 127;
val++;
val++;
val++;
System.out.println(val);
```

Casting

- What is print to the screen for vI1 and vI2?

```
double vD1 = 127.4;
int vI1 = (int) vD1;
System.out.println(vI1);


double vD2 = 127.6;
int vI2 = (int) vD2;
System.out.println(vI2);
```

Classes and Equality

- What is the result of the print statements?

```
String str1 = new String("abcabc");
String str2 = new String("abcabc");
String str3 = str1;
System.out.println(str1 == str2);
System.out.println(str3 == str1);
System.out.println(str3 == str2);
System.out.println(str1.equals(str2));
System.out.println(str1.substring(0, 3) == str2.substring(3,6));
System.out.println(str1.substring(0, 3).equals(str2.substring(3,6)));
```

Boolean Expressions

- Fill in the following table

A	B	A && B	A || B
True	True		
True	False		
False	True		
False	False		

Boolean Expressions

- Rewrite the following Boolean expressions without parentheses

Negated Expression	Equivalent Expression
!(A < B)	
!(A <= B)	
!(A > B)	
!(A >= B)	
!(A == B)	
!(A != B)	
!(A && B)	
!(A || B)	

If Statements

- What would be the output of the following two pieces of code?

```java
int a = 3;
if (a > 0) {
        System.out.println("0");
}
else if (a > 1) {
        System.out.println("1");
}
else if (a > 2) {
        System.out.println("2");
}
```

```java
int a = 3;
if (a > 0) {
        System.out.println("0");
}
if (a > 1) {
        System.out.println("1");
}
if (a > 2) {
        System.out.println("2");
}
```

If Statements

- What would be the output of the following two pieces of code?

```java
boolean a = false;
int[] arr = {0,1};
if (a && (arr[2] > 1)) {
      System.out.println("HI");
}
System.out.println("BYE");
```

```java
boolean a = false;
int[] arr = {0,1};
if ((arr[2] > 1) && a) {
      System.out.println("HI");
}
System.out.println("BYE");
```

If Statements

- What would be the output of the following two pieces of code?

```java
boolean a = true;
int[] arr = {0,1};
if (a || (arr[2] > 1)) {
        System.out.println("HI");
}
System.out.println("BYE");
```

```java
boolean a = true;
int[] arr = {0,1};
if ((arr[2] > 1) || a) {
        System.out.println("HI");
}
System.out.println("BYE");
```

Arrays

- What is the output of the following two snippets of code?

```
int[] arr1 = {0,1,2};
int[] arr2 = {0,1,2};
System.out.println(arr1 == arr2);
System.out.println(arr1[0] == arr2[0]);
```

```
String[] arr1 = {new String("str0"), new String("str1"), new String("str2")};
String[] arr2 = {new String("str0"), new String("str1"), new String("str2")};
System.out.println(arr1 == arr2);
System.out.println(arr1[0] == arr2[0]);
```

Pseudocode

- Write pseudocode for finding the largest value in an integer array
- Write pseudocode for finding the index of a target value in an integer array
- Write pseudocode for sorting an integer array

Pseudocode

- Write code that returns true if
  - (target_value - tol) <= value <= (target_value + tol)

```
private boolean within_tol(double value, double target_value, double tol) {
		//Implement
}
```

Pseudocode

- Write pseudocode to swap two values

Methods

- What are outputs of the two snippets of code?

```java
public class Test {
    public static void main(String[] args) {
        Test t = new Test();
        t.start();
    }
    public void start() {
        System.out.println("Start");
        printOne();
    }
    public void printOne() {
        System.out.println("One");
        printTwo();
    }
    public void printTwo() {
        System.out.println("Two");
        printThree();
    }
    public void printThree() {
        System.out.println("Three");
    }
}
```

```java
public class Test {
    public static void main(String[] args) {
        Test t = new Test();
        t.start();
    }
    public void start() {
        printOne();
        System.out.println("Start");
    }
    public void printOne() {
        printTwo();
        System.out.println("One");
    }
    public void printTwo() {
        printThree();
        System.out.println("Two");
    }
    public void printThree() {
        System.out.println("Three");
    }
}
```

Methods

- What are outputs of the two snippets of code?

```java
public class Test {
    public static void main(String[] args) {
        Test t = new Test();
        t.start();
    }
    public void start() {
        printOne();
        printOne();
        System.out.println("Start");
    }
    public void printOne() {
        printTwo();
        printTwo();
        System.out.println("One");
    }
    public void printTwo() {
        printThree();
        printThree();
        System.out.println("Two");
    }
    public void printThree() {
        System.out.println("Three");
    }
}
```

```java
public class Test {
    public static void main(String[] args) {
        Test t = new Test();
        t.start();
    }
    public void start() {
        printOne();
        System.out.println("Start");
        printOne();
    }
    public void printOne() {
        printTwo();
        System.out.println("One");
        printTwo();
    }
    public void printTwo() {
        printThree();
        System.out.println("Two");
        printThree();
    }
    public void printThree() {
        System.out.println("Three");
    }
}
```

Classes

- What is the output of the following method

```java
public class Dog {
    private double weight;
    public static void eat(Dog dog) {
        dog.weight++;
    }


    public Dog(double weight) {
        this.weight = weight;
    }


    public static void main(String[] args) {
        Dog dog1 = new Dog(1.0);
        Dog dog2 = dog1;
        System.out.println(dog1.weight);
        System.out.println(dog2.weight);

        Dog.eat(dog1);

        System.out.println(dog1.weight);
        System.out.println(dog2.weight);
    }
}
```

```java
public class Dog {
    private double weight;
    public static Dog eat(Dog dog) {
        return new Dog(dog.weight + 1.0);
    }


    public Dog(double weight) {
        this.weight = weight;
    }


    public static void main(String[] args) {
        Dog dog1 = new Dog(1.0);
        Dog dog2 = dog1;
        System.out.println(dog1.weight);
        System.out.println(dog2.weight);

        dog1 = Dog.eat(dog1);

        System.out.println(dog1.weight);
        System.out.println(dog2.weight);
    }
}
```

Mutability

- What problems could arise with getLegLengths() and how can they be resolved?

```
public class Dog {
    private double[] legLengths;

    public Dog(double[] legLengths) {
        this.legLengths = legLengths;
    }

    public double[] getLegLengths() {
        return this.legLengths;
    }
}
```

Inheritance and Polymorphism

- For overloaded methods, Java determines which method to call at _________.
  - Compile time
- Over overridden methods, Java determines which method to call at ________.
  - Run time

Inheritance and Polymorphism

- What is the output of the following code?

```
public class Animal {
    protected String name;
    protected double weight;
    public Animal(String name, double weight) {
        this.name = name;
        this.weight = weight;
    }
    public boolean equals(Animal animal) {
        return this.name.equals(animal.name) &&
        (this.weight == animal.weight);
    }
}
```

```
public class Dog extends Animal {
    private String earType;
    public Dog(String name, double weight, String
    earType) {
        super(name, weight);
        this.earType = earType;
    }
    public boolean equals(Dog dog) {
        return this.name.equals(dog.name) &&
        (this.weight == dog.weight) &&
        (this.earType.equals(dog.earType));
    }
}
```

```
public static void main(String[] args) {
    Animal dog1 = new Dog("Spike", 1.0, "Pointy");
    Dog dog2 = new Dog("Spike", 1.0, "Floppy");
    Dog dog3 = new Dog("Spike", 1.0, "Pointy");
    System.out.println(dog1.equals(dog2));
    System.out.println(dog1.equals(dog3));
    System.out.println(dog2.equals(dog3));
}
```

Inheritance and Polymorphism

- What is the output of the following code?

```
public class Animal {
    protected String name;
    protected double weight;
    public Animal(String name, double weight) {
        this.name = name;
        this.weight = weight;
    }
    public boolean equals(Animal animal) {
        return this.name.equals(animal.name) &&
        (this.weight == animal.weight);
    }
}
```

```
public static void main(String[] args) {
    Animal dog1 = new Dog("Spike", 1.0, "Pointy");
    Dog dog2 = new Dog("Spike", 1.0, "Floppy");
    Dog dog3 = new Dog("Spike", 1.0, "Pointy");
    System.out.println(dog1.equals(dog2));
    System.out.println(dog1.equals(dog3));
    System.out.println(dog2.equals(dog3));
}
```

```
public class Dog extends Animal {
    private String earType;
    public Dog(String name, double weight, String
    earType) {
        super(name, weight);
        this.earType = earType;
    }
    public boolean equals(Animal animal) {
        if (animal instanceof Dog) {
            Dog dog = (Dog) animal;
            return this.name.equals(dog.name) &&
            (this.weight == dog.weight) &&
            (this.earType.equals(dog.earType));
        } else {
                return super.equals(animal);
        }
    }
}
```