

Shortcuts & Review

COMP163

“It’s not that I’m so smart, it’s just that I stay with problems longer.”

Albert Einstein

First Exam

- The first exam in COMP163 will be in lecture on Monday, September 23
- The exam will cover everything since the beginning of the semester
- You are allowed one 8½ by 11 inch page of notes
- Most questions will be of the form
 - complete this program
 - what does this program display

Lab Quiz

- There will be a quiz in lab this week, Thursday, September 19, **tomorrow**
- A lab quiz is similar to a regular lab, but you must do it all by yourself
- You may use your notes, textbook, previous programs and the web
- You must be in the lab during your lab period to get credit for the quiz
- Lab quizzes count for 5% of your total grade, each quiz is about 1% of your course grade

Technical Interview Prep Sessions

- Tuesday, September 24 at 4:00-5:30PM in McNair 132
- Target audience is first semester freshman CS majors who are taking COMP163
- Will give hands-on practice at the kind of technical problem solving and coding questions that are given during Google interviews
- Demystify the interview process
- Share what I and other Google interviewers have looked for over the last decade
- Led by Dr. Dave Foulser of



Increment Operator

- You can increment or add one to an integer by putting ++ after the variable

```
cow++;
```

- This is the same as

```
cow = cow + 1;
```

- Example:

```
int bull = 5;
```

```
bull++;
```

```
System.out.println( bull ); // shows 6
```

Increment in Expression

- You can increment a variable in an expression

```
int cat = 3, dog = 5, bird = 7;  
cat = dog + bird++;
```

- **cat** has the value 12, **bird** has the value 8
- Post increment (with the ++ after the variable) logically occurs **after** the rest of the statement
- You cannot use ++ on an expression

```
dog = (cat + mouse)++; // error
```

Decrement Operator

- You can decrement an integer similarly by putting -- after the variable

```
COW--;
```

- You can decrement a variable in an expression

```
int  cat = 3, dog = 5, bird = 7;  
cat = dog + bird--;
```

- **cat** has the value 12, **bird** has the value 6.
- Post decrement (with the -- **after** the variable) logically occurs after the rest of the statement

```
cat = dog + bird;  
bird = bird - 1;
```


Pre-Increment and Decrement

- You can also increment or decrement a variable by putting ++ or -- before the variable

```
++frog;    --lizard;
```

- Pre-increment and decrement can also be used in an expression

```
int    cat = 3, dog = 5, bird = 7;  
cat = dog + ++bird;
```

- **cat** has the value 13, **bird** has the value 8.
- Pre increment (with the ++ **before** the variable) logically occurs before the rest of the statement

What is displayed?

```
int deer = 3, elk = 6, moose;  
moose = deer * ++elk;  
System.out.println( moose + " " + elk );
```

A. 18 6

B. 18 7

C. 21 6

D. 21 7

E. 24 7

What is displayed?

```
int deer = 3, elk = 6, moose;  
moose = deer * elk++;  
System.out.println( moose + " " + elk );
```

A. 18 6

B. 18 7

C. 21 6

D. 21 7

E. 24 7

Assignment Operators

- Often you will do arithmetic on a variable and store the result back in the same variable
- Assignment operators simplify arithmetic modification to a variable

```
dog = dog + 5;
```

- can be expressed as

```
dog += 5;
```

Assignment Operators

operator	example	same as
<code>+=</code>	<code>cat += dog + 2;</code>	<code>cat = cat + (dog + 2);</code>
<code>-=</code>	<code>cat -= dog + 2;</code>	<code>cat = cat - (dog + 2);</code>
<code>*=</code>	<code>cat *= dog + 2;</code>	<code>cat = cat * (dog + 2);</code>
<code>/=</code>	<code>cat /= dog + 2;</code>	<code>cat = cat / (dog + 2);</code>

Assignment Done Last

- Any equation to the right of an assignment operator is done before the assignment operation

```
cat *= dog + 5;
```

is equivalent to

```
cat = cat * ( dog + 5 );
```

Many Ways To Do the Same Thing

- To add one to an integer you can

```
dragon = dragon + 1;
```

```
dragon += 1;
```

```
dragon++;
```

```
++dragon;
```

- All will accomplish the task

What is displayed?

```
int grisly = 2, kodiak = 4;  
kodiak += grisly;  
System.out.println( kodiak + " " + grisly );
```

- A. 2 2
- B. 4 2
- C. 6 2
- D. 6 3
- E. none of the above

Exam Topics

- All of the material in Chapters 1 through 4 of the online textbook

Topics not covered in class
will not be on the exam

A word from our lawyer:

- The exam may contain questions from any of the material covered in class since the beginning of the class

One Page of Notes

- You are allowed one and only one 8½ by 11 inch page of notes during this exam
- You are not allowed to use more than 187 square inches of paper surface
- You will do better if you make your own page of notes and not copy your friend's notes

Exam Format

- The exam will be similar to the labs, quizzes and homework
- Exams tend to be programming oriented
- The exam will be similar to the practice exam available on Blackboard
- Try the sample exam before looking at the answers

Expressions

- Remember to use parenthesis to change the default order of execution
- The Math class can be used for most trigonometric and other math functions

Math Examples

$$distance = \sqrt{x^2 + y^2}$$

```
distance = Math.sqrt( x*x + y*y );
```

$$rate = b * \log_{10} \left(\frac{signal}{noise} + 1 \right)$$

```
rate = b * Math.log10( signal / noise + 1.0 );
```

Operator Priority

- When an expression has multiple operators the high priority operations are done first
 - When operators are of equal priority, they are executed left to right
1. Casting such as (`int`) or (`double`)
 2. pre-increment or decrement such as `++dog`
 3. `*` / `%`
 4. `+` -
 5. post-increment or decrement such as `dog++`

Evaluation of Expression

dog = 2 + 5 * 3 / 2;

dog = 2 + 15 / 2;

dog = 2 + 7;

dog = 9;

Evaluation of Expression

dog = (2 + 5) * 3 / 2;

dog = 7 * 3 / 2;

dog = 21 / 2;

dog = 10;

Evaluation of Expression

dog = (2 + (double)5) * 3 / 2;

dog = (2 + 5.0) * 3 / 2;

dog = 7.0 * 3 / 2;

dog = 21.0 / 2;

dog = 10.5;

Evaluation of Expression

```
int cat = 2;
```

```
dog = (2 + 5) * 3 / ++cat;
```

```
dog = 7 * 3 / ++cat;
```

```
dog = 7 * 3 / 3;
```

```
dog = 21 / 3;
```

```
dog = 7;
```

Which expression will generate a compiler error?

```
int cobra = 7;
```

```
double mamba = 1.1;
```

- A. cobra = mamba;
- B. mamba = cobra;
- C. all of the above
- D. none of the above

Integer Division

- Integers can only hold whole numbers
- If division results in a fractional part, the fraction is dropped and the result is just the whole number

$$8/2 \text{ is } 4$$

$$8 / 3 \text{ is } 2$$

$$1 / 2 \text{ is } 0$$

$$5 / 4 \text{ is } 1$$

Double Constants

- When you are writing an equation involving double type variables, always use constants with a decimal point

```
double dog = 2.0, cat;
```

```
cat = 1 / 2 * dog + 5; // does not work
```

```
cat = 1.0 / 2.0 * dog + 5.0; // correct
```

Organizing a Program

- The general flow of a computer program is:
 - Read some input data
 - Calculate a value from the input
 - Display the results
- Some programs will repeat this process many times

Which Program Segment is Correct?

A

```
dog = cat * 5;  
System.out.print("enter >");  
cat = keyboard.nextInt();  
System.out.println(dog);
```

B

```
System.out.print("enter >");  
dog = cat * 5;  
System.out.println(dog);  
cat = keyboard.nextInt();
```

C

```
System.out.print("enter >");  
cat = keyboard.nextInt();  
dog = cat * 5;  
System.out.println(dog);
```

D

```
System.out.print("enter >");  
System.out.println(dog);  
dog = cat * 5;  
cat = keyboard.nextInt();
```


Calling Methods of Classes

- Static methods are called on a class

classname.method()

- `Math.cos(dog)` is a static method

Calling Methods of Objects

- Non-static methods (sometimes called instance methods) are called on objects, not classes

```
Scanner keyboard = new Scanner(System.in);
```

```
double dog = keyboard.nextDouble();
```

```
JTextField rat = new JTextField();
```

```
String mouse = rat.getText();
```

Methods inside a Class

class

main Method

cube Method

Methods in Any Order

class

cube Method

main Method

What is displayed?

```
public class Colors {  
    static void aqua() {  
        grass();  
        System.out.println("Blue");  
    }  
    static void grass(){  
        System.out.println("Green");  
    }  
    public static void main(String[] args) {  
        System.out.println("Gray");  
        aqua();  
    }  
    static void rouge() {  
        System.out.println("Scarlet");  
    }  
}
```

- A. Blue
Green
Gray
Scarlet
- B. Gray
Blue
- C. Scarlet
Blue
Gray
- D. Gray
Green
Blue
- E. none of the above

Parameter Values Copied

```
public class PassParm {  
    public static void main(String[] args ) {  
        int    cat = 5, dog = 7, bird = 3;  
        bird = example( cat, dog );  
        System.out.println("main "+bird);  
    }  
  
    static int example( int cow, int bull ) {  
        System.out.println("example "+cow);  
        return cow + bull;  
    }  
}
```

Displays

example 5

main 12

static and non-static Method Calls

- Static methods are called on the class, not a specific object
- `Math.sqrt()` and `System.in` are static
- non-static methods are called on an object of the class, not the class itself

```
java.util.Scanner keyboard =  
    new java.util.Scanner(System.in);  
myVar = keyboard.nextDouble();
```

String Methods

- There are many methods that can be called on objects of the String class
- `length()` Returns the length of this string

```
String lecture = "First exam";
```

```
int howLong = lecture.length(); // 10
```


Searching Strings

- You can search a string to see if it contains the another string
- `indexOf(String str)` Returns the index within this string of the first occurrence of the specified String or -1 if not found

indexOf Example

- The position of a character in a string starts counting at zero

```
String major = "Computer Science";  
              // 0123456789012345
```

```
int where = major.indexOf( "c" );    // 10
```

```
where = major.indexOf( "wrong" );   // -1
```

Taking Strings Apart

- `substring(int beginIndex, int endIndex)` Returns a new string that is a substring of this string. The substring begins at `beginIndex` and extends to the character at position **`endIndex - 1`**. Thus the length of the substring is `endIndex - beginIndex`
- `charAt(int index)` Returns the char value at the specified index

substring Example

```
String major = "Computer Engineering";  
              //              111111111111  
              //              01234567890123456789
```

```
String result = major.substring( 11, 14 );
```

result has the value "gin"

Interesting String Methods

- length
- indexOf
- substring
- charAt
- toUpperCase
- toLowerCase
- trim

Making a GUI

1. Create a class that extends JFrame and implements `java.awt.event.ActionListener`
2. Create component objects
3. Specify the layout manager
4. Add the component objects to the container
5. Add an action listener to the components that respond to user input
6. All the program to be visible and stop
7. Create an `actionPerformed` method

Usual Java GUI

```
public class MyProgram extends javax.swing.JFrame
    implements java.awt.event.ActionListener {
    // declare GUI components here
    public MyProgram() {
        // setup GUI here
    }
    public void actionPerformed(
        java.awt.event.ActionEvent dog){
        // program here
    }
    public static void main(String[] cat) {
        MyProgram fish = new MyProgram();
    }
}
```

Adding Components to a GUI

- To make them appear in a applet the components must be added to the frame
- The add method links a component to the content pane (variable goat)

```
goat.add( GUIthing1 );  
goat.add( GUIthing2 );
```

- The order in which the components are added determines the order in which they will appear in the GUI

Adding a Listener

- The method `addActionListener` should be called on each GUI component that can do something
- A button will not do anything unless you specify an action listener

```
javax.swing.JButton squirrel = new  
    javax.swing.JButton("Go");
```

...

```
squirrel.addActionListener( this );
```

```
JTextField bird = new JTextField();  
JTextField cow = new JTextField();
```

```
// Try A
```

```
String cat = bird.getText();  
double dog = Double.parseDouble(cat);  
cat = cow.getText();  
double goat = Double.parseDouble(cat);
```

```
// Try B
```

```
double dog = Double.parseDouble(cat);  
double goat = Double.parseDouble(bull);  
String cat = bird.getText();  
String bull = cow.getText();
```

Which is Correct?

A. A

B. B

C. both

D. neither

Graphics Methods

```
bird.drawRect( x, y, width, height );
```

```
bird.fillRect( x, y, width, height );
```

```
bird.drawOval( x, y, width, height );
```

```
bird.fillOval( x, y, width, height );
```

```
bird.drawLine( x1, y1, x2, y2 );
```

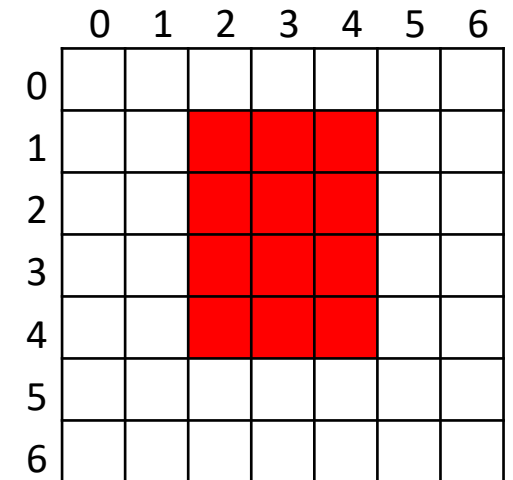
```
bird.drawString( String, x, y)
```

- where **bird** is the parameter to paint, an object of the type `java.awt.Graphics`

Work with your team

Complete this method to draw a small **red** rectangle as shown in the diagram. Each box represents a pixel

```
public void paint( java.awt.Graphics bird ) {
```

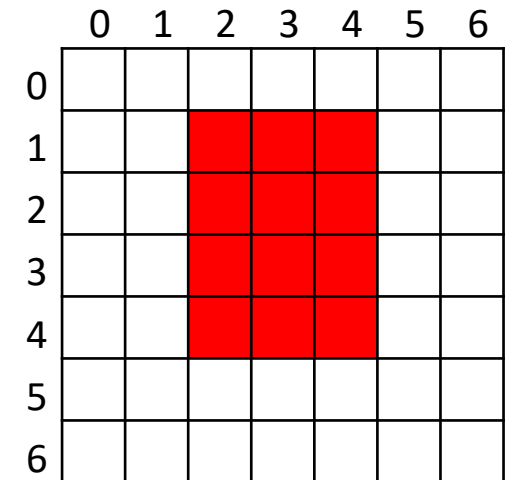


Possible Solution

Complete this method to draw a small **red** rectangle as shown in the diagram. Each box represents a pixel

```
public void paint( java.awt.Graphics bird ) {
```

```
    bird.setColor( java.awt.Color.RED );  
    bird.fillRect( 2, 1, 3, 4 );
```



Likely Exam Questions

- Convert math equations to Java
- Show what will be displayed by a program
- Complete parts of a program
- Write the actionPerformed method for a GUI

- Know the string methods
- Know the graphics methods

First Exam

- The first exam in COMP163 will be in lecture on Monday, September 23
- The exam will cover everything since the beginning of the semester
- You are allowed one 8½ by 11 inch page of notes
- Most questions will be of the form
 - complete this program
 - what does this program display

Lab Quiz

- There will be a quiz in lab this week, Thursday, September 19
- A lab quiz is similar to a regular lab, but you must do it all by yourself
- You may use your notes, textbook, previous programs and the web
- You must be in the lab during your lab period to get credit for the quiz
- Lab quizzes count for 5% of your total grade, each quiz is about 1% of your course grade

Technical Interview Prep Sessions

- Tuesday, September 24 at 4:00-5:30PM in McNair 132
- Target audience is first semester freshman CS majors who are taking COMP163
- Will give hands-on practice at the kind of technical problem solving and coding questions that are given during Google interviews
- Demystify the interview process
- Share what I and other Google interviewers have looked for over the last decade
- Led by Dr. Dave Foulser of 