1 Disclaimer

This is a review of the material so far, but there may be material on the exam not covered in this study guide.

2 Variables

2.1 Primitive Types

The eight primitive types are: int, double, char, float, long, short, byte, and boolean. So far you’ve learned about two reference types, a String and a Scanner. These are NOT a primitive types, you will learn why later in the semester.

Below are a few examples of declaring a variable. Format: type var_name;

```java
int i;
double d;
char c;
String s;
```

Below are a few examples of assigning a variable. Format: var_name = value;

```java
i = 3;
d = 4.5;
c = 'a';
s = "Hey all!";
```

Below are a few examples of initializing a variable. Initializing is the combination of declaring and assigning. Format: type var_name = value;

```java
boolean b = true;
float f = 3.1415F;
long l = 32L;
String response = "Yes";
```

Remember: Every variable must always have a declaration (assignment can come later).

Below are a few examples of typecasting.

```java
int i = 3;
double d = (double) i;
char c = 'c';
int asciiC = (int) c;
```

Remember: If you typecast you can lose information, for example if I typecast 6.8 to an integer I will get back 6.

2.2 Suggestions, Warnings, and Resources

- Warning: Make sure you use the correct ticks/quotes for variables. For example, to assign(initialize a char you must use ‘ ’ (ticks or single quotes), to assign(initialize a String you must use “ ” (double quotes), and numbers don’t use ticks or quotes.

2.3 FAQs

1. Q: Why is there an ‘F’ or ‘L’ at the end of a long or float value? Is it case sensitive?
   A: If you leave off the ‘F’ when assigning or initializing a float, Java will assume you want a double (instead of a float). A float can hold 32-bit number and a double can hold a 64-bit number. You only need to add an ‘L’ when assigning(initializing a long when the value won’t fit into an int. An int can hold numbers between $2^{-31}$ and $2^{31} - 1$ and a long can hold numbers between $2^{-63}$ and $2^{63} - 1$. The ‘F’ and the ‘L’ are not case sensitive, but be careful a lower case ‘l’ looks an awful like the number 1.
3 Printing

So far you know of two ways to print to the console: `System.out.print();`, `System.out.println();`. Below is an example of how and when you could use each.

```java
int num1 = 3;
int num2 = 5;
int num3 = 9;
double avg = (num1 + num2 + num3) / 3;
// The following three print statements are equivalent
System.out.println("Average \(\frac{\text{num1} + \text{num2} + \text{num3}}{3}\) : \(\text{avg}\)");
System.out.println("Average \(\text{num1} + \text{num2} + \text{num3}\) : \(\text{avg}\)");
```

This outputs the following:

Average 3, 5, and 9: 5.0
Average 3, 5, and 9: 5.0

- `System.out.print();` is the same as `System.out.println();` except it doesn’t have a new line character at the end of the line.

3.1 FAQs

1. Q: Can you use either print or println to print? Is there a specific time you’d use one over the other?
   A: You can use either to print at any time, however, one might be easier in some situations over the other.

4 Conditional Statements

4.1 if-else if-else Statements

```java
public class Conditionals {
    public static void main (String [] args){
        int i = 0;
        // if statement
        if (i == 0) {
            System.out.println("i == 0");
        }
        boolean b = true;
        // I don’t need brackets because I only have one line of code
        // under each statement. However, you can always have brackets
        // like the example above.
        if (b == false)
            System.out.println("b is FALSE");
        else
            System.out.println("b is TRUE");
        int i1 = 32;
        // here I do need brackets because there is more than one
        // line of code nested in my if statement.
        if (i1 >= 15){
            System.out.println("i1 is greater than or equal to 15");
            i1++;
        } else if (i1 < 3)
            System.out.println("i is less than 3");
        // elsede are like catch—alls there are a section of numbers
        // that aren’t covered by my if and else if so my else will
        // handle those values
        else
            System.out.println("i must be between 3 and 14");
        // sometimes you don’t need an else
        String name = "Waldo";
        // if I have found Waldo I want to say I found him but...
    }
}
```
if (name.equals("Waldo"))
    System.out.println("Found him!");
    // if I didn’t find him I don’t want to do anything
}

The output from the above code is:
i == 0
b is TRUE
i1 is greater than or equal to 15
Found him!

4.2 Difference between if-if-if and if-else if-else

If you use an if-else if-else statement and multiple conditionals are true, the program executes the first true
conditional and then exits.

If you use an if-if-if statement and multiple conditionals are true, the program will execute all of the true
conditionals.

The output from the above code is:

if-else if-else statement
i == 128
i > 100
i < 130

if-if-if statement
i == 128
i > 100
i < 130

// beware of your breaks!
char c = ‘b’;

4.3 Switch Statements

switch (i) {
case 10: System.out.println("i = 10"); break;
case 32: System.out.println("i = 32"); break;
case 101: System.out.println("i = 101"); break;
default: System.out.println("i is not 10, 32, or 101");
}
switch (c) {
    case 'a': System.out.println("c = a");
    case 'b': System.out.println("c = b");
    case 'c': System.out.println("c = c");
    default: System.out.println("c is not a, b, or c");
}

// times you might want to not have breaks
int dogs = 0, cats = 0;
String breed = "German Shepard";

// spacing doesn’t matter
// you aren’t required to have a default
// like you aren’t required to have an else
// just be aware that there are possible cases
// that aren’t being handled.
switch (breed) {
    case "Boxer":
    case "Great Dane":
    case "German Shepard":
    case "Lab":
        dogs++; break;
    case "Calico": case "British Shorthair": case "Siamese":
        cats++; break;
}
System.out.println("Dogs: "+ dogs + ", Cats "+ cats);

The output from the above code is:
i = 32
c = b
c = c
c is not a, b, or c
Dogs: 1, Cats 0

4.4 Suggestions, Warnings, and Resources

- **Warning:** Don’t add a semicolon after your conditional. For example if (32 == 32);
- **Warning:** Be aware of when you should and shouldn’t have breaks in your switch statement.

5 Scanners

Below is an example of reading from the keyboard/console.

```java
// importing necessary methods
import java.util.Scanner;

public class ScannerPractice {
    public static void main (String [] args) {
        // Initializes a Scanner object called reader
        Scanner reader = new Scanner (System.in);
        // Prompting user for an integer
        System.out.print("Enter your favorite integer: ");
        // Read and store an int
        int favorite_number = reader.nextInt();
        // Prompting user for a decimal
        System.out.print("Enter your favorite decimal number: ");
        // Read and store a double
        double favorite_decimal = reader.nextDouble();
        // printing summary
        System.out.println("Your favorite integer is "+ favorite_number + ", Your favorite decimal number is "+ favorite_decimal);
        reader.close();
    }
}
```
On the console, it looks like this:

Enter your favorite integer: 3
Enter your favorite decimal number: 3.14
Your favorite integer is 3
Your favorite decimal number is 3.14

5.1 Suggestions, Warnings, and Resources

- Suggestion: Get in the habit of closing your Scanners
- Resource: Use Oracle's documentation for more methods and examples

5.2 FAQs

1. Q: My code still works even though I didn’t close my Scanner object, why do you guys teach so you have to close it?
   A: We teach it so you close your Scanners so you get in the habit of closing objects (better coding practice and when we get to PrintWriters you must close your object).
6 Practice Written Exam

6.1 Short Answer

1. Declare an integer named `i0`.

2. Initialize a float named `f` to 4.0.

3. Initialize a Scanner to read from the keyboard, called `read`.

4. Assign `i0` to an integer entered from the user.

5. Close the Scanner.

6. Name the eight primitive types.

7. What is 18 % 3?

8. Write a switch statement based off of the character variable `c`. If `c` is ‘a’ print “hi”, if `c` is ‘b’ print “okay”, if `c` is ‘c’ print “then”, if `c` is ‘d’ print “bye”, if it is not ‘a’, ‘b’, ‘c’, or ‘d’ print “fine then”.

9. Write an if statement that does the same thing as question 8.

10. Declare a Scanner called `scan`. Read in one int and store it into the predefined int variable called `my_int`. Close your scanner.
6.2 Tracing Code

For each question below, write what is printed.

```java
import java.util.Scanner;

public class Tracing {
    public static void main(String[] args) {
        int i = 3;
        double d = (double) i;
        // Question 1
        System.out.println(i + " " + d);

        Scanner keys = new Scanner(System.in);
        d = keys.nextInt();
        i = (int) keys.nextDouble();
        // Question 2 (assume the user enters 8 and 98.5 respectively)
        System.out.println(i + " " + d);
        String result = "initial result";
        if (d > 98)
            result = "greater than 98";
        else
            result = "less than or equal to 98";
        System.out.println(result);

        char grade = 'a';
        // Question 3
        switch (grade) {
            case 'a': System.out.println("Passing!");
            case 'b': System.out.println("Passing!");
            case 'c': System.out.println("Passing!");
            case 'd': System.out.println("Kinda sorta passing");
            case 'f': System.out.println("Not really passing");
            default: System.out.println("Not a grade letter recognized");
        }
    }
}
```
7 Programming Quiz Practice Exam

1. Create a class with a main method
2. Import the Scanner class
3. Create an int variable called age
4. Prompt the user to enter their age
5. Using the Scanner read in their age and assign the value to age
6. Write a conditional statement that prints the following under these conditions:
   - if their age is 16 or older print “You can drive!”
   - if their age is 18 or older print “You can vote!”
   - if their age is 21 or older print “You can drink!”
   - if their age is 25 or older print “You can rent a car!”
7. Close reader

An example run would look like:
Enter your age: 25
You can drive!
You can vote!
You can drink!
You can rent a car!

8 Suggestions for Studying and Test Taking

8.1 Written

When reading through code and writing the output: Write your variables on the side and as your variables change, you change your list on the side.

Practice writing code in Eclipse and before you run your program guess what the output would be. This is good practice for testing your own programs and also for the code tracing part of the exam.

If you need more tracing examples (or more coding examples in general), there is a “Programs” tab on the CS150 homepage. There are also examples on the Progress page.

8.2 Programming Quiz

Go back to past recitations and assignments and redo them without using the INTERNET, friends, or past recitations/assignments.

Practice writing code in Eclipse. Make up projects and problems or ask a TA and they can give you some challenges.

Look at code, the more exposure you get to code (whether it’s your own code or not) the easier it is to understand. Some sample code is under the “Programs” tab and the Progress Page.

8.3 Common Errors

- Incorrect brackets around conditional statements

8.4 Challenges

CodingBat and Hackerrank offer good extra coding practice.
ANSWERS ON THE NEXT PAGE
9 Answers to Practice Written and Programming Problems

9.1 Written

9.1.1 Short Answer

1. int i0;
2. float f = 4.0F;
3. Scanner read = new Scanner(System.in);
4. i0 = read.nextInt();
5. read.close();
6. int, double, char, float, long, boolean, byte
7. 0
8. switch (c) {
   case 'a': System.out.println("hi"); break;
   case 'b': System.out.println("okay"); break;
   case 'c': System.out.println("then"); break;
   case 'd': System.out.println("bye"); break;
   default: System.out.println("fine then"); break;
}
9. if (c == 'a')
   System.out.println("hi");
else if (c == 'b')
   System.out.println("okay");
else if (c == 'c')
   System.out.println("then");
else if (c == 'd')
   System.out.println("bye");
else
   System.out.println("fine then");
10. Scanner scan = new Scanner (System.in);
    my_int = scan.nextInt();
    scan.close();

9.1.2 Tracing

1. 3 3.0
2. 98 8.0
3. less than or equal to 98
4. Passing!
   Passing!
   Passing!
   Kinda sorta passing
   Not really passing
   Not a grade letter recognized
9.2 Programming

NOTE: Your solution may be different from the code below, as long as your output matches my output (in all cases), you are answer is fine.

```java
import java.util.Scanner;

public class PracticeProgramming {
    public static void main(String[] args) {
        Scanner reader = new Scanner(System.in);
        int age;

        System.out.print("Enter your age:");
        age = reader.nextInt();

        if (age >= 16)
            System.out.println("You can drive!");
        if (age >= 18)
            System.out.println("You can vote!");
        if (age >= 21)
            System.out.println("You can drink!");
        if (age >= 25)
            System.out.println("You can rent a car!");

        reader.close();
    }
}
```