CS 160 - Exam 1 Study Guide and Practice Exam

July 1, 2016

Summary

1 Disclaimer

2 Variables
   2.1 Primitive Types
      2.1.1 Printing
   2.2 Reference Types

3 Loops
   3.1 For loops
   3.2 While loops
   3.3 Do-While Loops
   3.4 Hints and Warnings

4 Methods
   4.1 Return Values
   4.2 Parameters

5 Discrete Math
   5.1 Propositional Logic

6 Practice Written Exam
   6.1 Short Answer
   6.2 Tracing Code

7 Programming Quiz Practice Exam

8 Suggestions for Studying and Test Taking
   8.1 Written
   8.2 Programming Quiz
   8.3 Common Errors
   8.4 Challenges
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

//Declaration
int i;
double d;
char c0, c1, c2;

//Assigning
i = 3;
d = 4.0;
c0 = 'a';
c1 = 'b';
c2 = 'c';

//Initialization
int myInt = 0;
double myDouble = 1.0;
char myChar = 'd';
float myFloat = 2.2F;
long myLong = 1.23L;

//Type casting
double d;
int i = 3;
char c = 'a';
d = (double) i;
System.out.println((int)c);

2.1.1 Printing

//Printing with a new line
System.out.println("hi");
System.out.print("hi
");
System.out.printf("%s\n", "hi");

//Printf formats
int i = 0;
double d = 3.22311090;
System.out.printf("%d\n", i); //prints 0
System.out.printf("%4d\n", i); //prints with 3 spaces then 0
//default 6 decimal points and it rounds! (3.223111)
System.out.printf("%f\n", d);
System.out.printf("%.3f\n", d); //prints 3.223
2.2 Reference Types

```java
public class Review {
    public static void main (String [] args))
    {
        //Declaration
        String s;
        //here is an example of creating an object of the class
        Review r = new Review();

        //Assigning
        s = "Hi there";

        //Declaring & Assigning
        String myString = "Okay then";
        String myString1 = new String ("Okay fine then");

        //Common Methods
        String s = "SmIlEy m0nKEy";
        char c0 = s.toCharArray(); //assigns S to variable c
        int i0 = s.indexOf('z'); //assigns -1 to variable i0
        //because there is no 'z' in s
        String sub = s.substring(0); //assigns s to sub
        // returns
        // (stringName.substring(#)
        // the character from #
        // to the end of the string
        String s1 = s.substring(0,1); //only returns S because
        //stringName.substring(inclusive index, exclusive index)
        System.out.println(s.toUpperCase());
        String newString = s.toLowerCase();
        System.out.println(s.equals(newString)); //prints false
        boolean b = s.equals(sub); //assigns true to b
        int length = sub.length(); //assigns 13 to length
        // (1 based not 0 based
        // (like indexes)
    }
}
```

3 Loops

3.1 For loops

For loops are used when you know how many iterations the loop needs to complete.
General syntax:
```java
for (int i = 0; i < someLength; i++){
    //some code
}
```

You can also think of for loops as such:
```java
for (initialization; range; update) {
    //some code
}
```

"Real life example": A loop used print every second letter backwards (z, x, v, t...).
```java
for (char c = 'z'; c >= 'a'; c-=2){
    //some code
}
```
3.2 While loops

While loops are generally used when you don’t know how many iterations you need the loop to complete (i.e. if you are reading entered words from the keyboard until the user types “stop”).

General syntax:

```java
int i = 0;
while (i < someLength){
    // some code
    i++;}
```

You can also think of for loops as such:

```java
initialization; //outside of loop
while (range){
    //some code
    update
}
```

Same “Real life example” as above.

```java
char c = 'z';
while (c >= 'a'){
    System.out.print(c);
    c-=2;
}
```

3.3 Do-While Loops

Do-While loops are used when you want the loop to run at least once, regardless of whether the value is in the range. General syntax:

```java
int i = 0;
do {
    //code
    i++;
} while (range);
```

Same “Real life example” as from before.

```java
char c = 'z';
do {
    System.out.print(c);
    c-=2;
} while (c >= 'a');
```

3.4 Hints and Warnings

- Warning: Watch the ranges! When a String’s length is of size 8, i < 9 is the same as i ≤ 8. However, i ≤ 9 will give you an error.

- Hint: You can always change your update by using the “+=” or “-=” method, especially if you are updating by more than 1.
• Hint: You can always change your starting point, you don’t have to start at 0, it is just the most common. If you were asked to reverse a string, you can always make the starting point at stringName.length-1! (Remember the -1 though, lengths are 1 based while loops are 0 based usually).

• Summary: Basically you can solve a problem with loops in many different ways whether it’s using a different loop, starting at a different place, changing your ending value, or changing how you update.

4 Methods

4.1 Return Values

When a method has a return value of void it returns no value. You can print or manipulate instance variables and objects, without using the return keyword.

When a method has a return value of double, int, String, etc you must return a value of that type. For example, public int multiply(int a, int b) must return an integer. Example of a void method could be public void printMe(String s).

4.2 Parameters

Parameters are values that are passed in through the method when it is called. These are variables that you can use and manipulate (BEWARE of pass-by-reference(Objects) and pass-by-value(primitive types)!).

Given the following code, notice that we passed the multiply method myInt and myDouble but we used the variables i and d. This is where the warning comes in, this does not change any value. In order for the any change to be seen in the main method we would need to either assign to a variable or print.

```java
public static void main (String [] args0){
    int myInt = 3;
    double myDouble = 4.0;
    // will this do anything?
    multiply(myInt, myDouble);
}

public static double multiply(int i, double d){
    double result = i * d;
    return result;
}
```

5 Discrete Math

5.1 Propositional Logic

This is not comprehensive.

• Tautology: a proposition that is always true

• Contradiction: a proposition that is always false

• Contingency: a proposition that is neither always true and neither false (aka neither a tautology nor a contradiction)

• Given p → q
  Converse: q → p - Reverse the order
  Inverse: ¬ p → ¬ q - Negate the original order
  Contrapositive: ¬ q → ¬ p - Negate and reverse the order.
6 Practice Written Exam

If you would like practice in Discrete Math: go to the CS 160 Homepage and under the Tutorials tab there are practice quizzes there.

6.1 Short Answer

1. Declare an integer name i0.
2. Declare and assign 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. Change the pre-assigned String variable called s to be all upper case.
7. Name four primitive types.
8. Is Math.floor() one-to-one? Strictly increasing/decreasing?
9. Given that String s = new String ("Bob Marley"); String m = "mummies"); find the following (write “error” if necessary, assume these are printed using println):
   1. m.substring(0,3)
   2. s.substring(4)
   3. s.substring(0,11)
   4. m.charAt(1)
   5. s.charAt(1)
   6. s.charAt(11)
   7. s.indexOf('q')
   8. m.indexOf(3)
   9. s.length()
   10. s + m
   11. m.indexOf('m')
   12. s.length() + m.length()
10. What is 18 % 3?
11. Write a for loop, while loop, and do-while loop that prints odd numbers from 1 to 101 all on one line.
12. 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”.
13. Write an if statement that does the same thing as question 12.
14. Declare a Scanner called scan. Read in one word and store it into the predefined String variable called word. Print out the second through fourth character of word. Close your scanner.
15. Check to see if the predefined String variables s0 and s1 are equal. Print the result.
16. Check to see if the predefined int variables i0 and i1 are equal. Print the result.
17. List the name of the method, the return value, and the parameters.
   1. public static double multiply (int i)
   2. public String add (String s, char c, int i)
   3. public static int divide (int i, double d)
   4. public void printMe ()
6.2 Tracing Code

/* Instructions:
 * By each commented number write down what the console would print.
 * There are some questions that return errors. Either write "error" or you can write the exception name.
 * If you find an infinite loop, continue through the questions (even though the rest of the program would never print).
 * Write "infinite loop" for those cases for the questions.
 */
import java.util.Scanner;
public class Trace {
    public static void main (String [] args){
        char c = 'b';
        String s0 = "Hello World"
        // Question 1: What does this print?
        switch (c){
            case 'a':
                System.out.println("c = a");
            case 'b':
                System.out.println("c = b");
            case 'c':
                System.out.println("c = c");
            case 'd':
                System.out.println("c = d");
            default:
                System.out.println("c is not a, b, c, or d");
        }
        // Question 2: What is printed?
        s0.toLowerCase();
        if (s0.equals("hello world"))
            System.out.println("hiya Mars");
        else
            System.out.println("s0 doesn’t say hello world...");
        // Question 3: What does the console print EXACTLY
        for (int i = 0; i < 30; i++){
            if (i % 3 == 0){
                System.out.print(i + ", ");
            }
        }
        System.out.println(); // used for spacing
        int userNum = 3;
        // Question 4:
        do {
            System.out.println("The userNum value is: " + userNum);
        } while (userNum > 5);
        // Question 5:
        for (int i = 0; i < s0.length(); i++)
            System.out.print(s0.charAt(i) + ":");
        System.out.println(); // used for spacing
        // Question 6:
        System.out.println(s0.indexOf('u'));
//Question 7:
for (int i = 1; i > 0; i++)
    if (i % 2 == 1)
        System.out.print(i + ";");
System.out.println(); // used for spacing

7 Programming Quiz Practice Exam

Go to www.hackerrank.com/midterm-practice. There are a ton of programming questions that will help you prepare for the programming quiz.

8 Suggestions for Studying and Test Taking

8.1 Written

There are times when > and ≥ cause the same result (for example: int age = 8; if (age > 7) and if (age ≥8)).

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.

Write down for, while, do-while loops/templates. Know what each piece does, that way when you come across a problem that needs a loop then you can quickly change a piece (initialization/starting point, range, or the update) of the loop to solve your problem.

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.

For your discrete math there are interactive quizzes under the Tutorials tab on the CS 160 Homepage.

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. Under the Progress Page many of the units have programs that you can look at.

8.3 Common Errors

• Loop ranges either go too far or too little
• Scanner token processing(.next, .nextInt(), .nextDouble()) to line processing (.nextLine()). Need an extra .nextLine() when going from token processing to line processing.
• Use .equals when comparing Strings, NOT == !
• Incorrect brackets around loops and conditional statements

8.4 Challenges

Under the Tutorials tab on the CS 160 Homepage there are practice Discrete Math Quizzes. Some coding practice sites, like CodingBat.com or Hackerrank.com are also very useful.