CS 150 - Exam 2 Study Guide and Practice Exam

October 9, 2017

Summary

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 printf

`System.out.printf();` uses format specifiers as placeholders. A format specifier begins with a percent sign and ends with a conversion symbol.

```
%d - int
%f - float
%.#f - (# is the number of decimal places) float
%c - char
%s - String
```

2.1 Examples

```
public class PrintfExamples {
  public static void main (String [] args) {
    int i = 89;
    char c = 'a';
    String s = "'Sup bro!";

    System.out.printf("This is my integer: %d\n", i);
    System.out.printf("This is my character: %c\n", c);
    System.out.printf("This is my String: %s\n", s);
    System.out.printf("This is PI to six decimal places: %f\n", Math.PI);
    System.out.printf("This is PI to three decimal places: %.3f\n", Math.PI);
    System.out.printf("This is PI to nine decimal places: %.9f\n", Math.PI);

    double d = 1.239;
    System.out.printf("Printf rounds! d to six decimals %f, d to two decimals %.2f\n", d, d);
    System.out.printf("Printf pads! d to 19 decimals %.19f\n", d);
    System.out.printf("Printf is nice for printing multiple variables:
                      %d, %c, %s, %.4f, %.6f\n", i, c, s, Math.PI, d);
  }
}
```

Output from above code:

```
This is my integer: 89
This is my character: a
This is my String: 'Sup bro!
This is PI to six decimal places: 3.141593
This is PI to three decimal places: 3.142
This is PI to nine decimal places: 3.141592654
Printf rounds! d to six decimals 1.239000, d to two decimals 1.24
Printf pads! d to 19 decimals 1.2390000000000000000
Printf is nice for printing multiple variables: 89, a, 'Sup bro!, 3.1416, 1.239000
```

2.2 Suggestions, Warnings, and Resources

- Resource: Here is a good printf resource.
- Resource: Oracle's Formatter Documentation
- Remember: If you don't specify how many decimals to print when using printf, Java defaults to printing six decimal digits.
- Warning: Printf rounds!

2.3 FAQs

1. **Q:** Is it necessary to use printf or String.format if you need to specify decimal places?
   A: Yes.

3 Strings

Common String methods:

- length (ex. `s.length();`)
- charAt (ex. `s.charAt(0);`)
- indexOf (ex. `s.indexOf('a');`)
- substring (ex. `s.substring(0, 3);`)
- equals (ex. `s.equals(s1);`)
- toUpperCase (ex. `s.toUpperCase();`)
- toLowerCase (ex. `s.toLowerCase();`)

Below are a few examples of how to use the above methods.

```
public class StringExamples {
   public static void main (String [] args) {
         String s0 = "SmIlEy mOnKEy";
         char c0 = s0.charAt(0);   //assigns S to variable c
         System.out.println("c0: " + c0);

         int i0 = s0.indexOf('z'); //assigns -1 to variable i0
                                   //because there is no 'z' in s
         System.out.println("i0: " + i0);

         //stringName.substring(inclusive index, exclusive index)
         String s1 = s0.substring(0); //assigns s0 to s1
                                      // returns
                                      // (stringName.substring(#)
                                      //the character from #
                                      //to the end of the string
         System.out.println("s1: " + s1);

         String s2 = s0.substring(0,1); // Remember: inclusive index as the first parameter
    and
                                        // an exclusive index for the second.
                                        // Meaning, this would only read the first letter.
         System.out.println("s2: " + s2);

         System.out.println(s0.toUpperCase()); // does not change s!
         String newString = s0.toLowerCase();  // does not change s!
         System.out.println("newString: " + newString);

         System.out.println(s0.equals(newString));
         boolean b = s0.equals(s1);
         System.out.println("b: " + b);
         System.out.println(s0 == newString);

         int length = s1.length(); //(1 based not 0 based
                                   //(like indexes)
         System.out.println("length: " + length);
    }
}
```

Output from the above code:

```
c0: S
i0: -1
s1: SmIlEy mOnKEy
s2: S
SMILEY MONKEY
newString: smiley monkey
false
b: true
false
length: 13
```

3.1 Suggestions, Warnings, and Resources

- Resource: Use Oracle's documentation for more methods and examples
- Warning: Always use String's `.equals` method (compared to `==`).
- Warning: Indexing for Strings always starts at 0 (for example the first character is the same as the 0th character).
- Warning: String methods do NOT change the original string!

4 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();
    // Prompting user for a character
    System.out.print("Enter your favorite character: " );
    // Read and store a letter
    char favorite_character = reader.next().charAt(0);

  // printing summary
  System.out.printf("Your favorite integer is %d\nYour favorite decimal number to 6 decimals
    is %f\nYour favorite character is %c\n", favorite_number, favorite_decimal,
    favorite_character);
  }
}
```

On the console, it looks like this:

```
Enter your favorite integer: 11
Enter your favorite decimal number: 3.14
Enter your favorite character: a
Your favorite integer is 11
Your favorite decimal number to 6 decimals is 3.140000
Your favorite character is a
```

4.1 Suggestions, Warnings, and Resources

- Suggestion: Get in the habit of closing your Scanners
- Warning: When you go from line based processing (.nextLine()) to token based processing (.next(), .nextDouble(), .nextInt()) you MUST discard the new line character before you continue parsing.
- Resource: Use Oracle's documentation for more methods and examples

4.2 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).

5 Loops

5.1 for loops

For loops are generally used when you know when you want to stop. For example if you need to count to 100 or if you need to loop the length of a String.

General syntax:

```
for (initialize; termination condition; update) {
  // code
}
```

A few examples:

```
public class ForLoops {
  public static void main (String [] args) {
    for (int i = 0; i < 10; i++)
      System.out.print(i + " ");
    System.out.println(); // used for spacing

    String s = "Hello! How are you?";
    for (int i = 0; i < s.length(); i++)
      System.out.print(s.charAt(i) + ":");
    System.out.println(); // used for spacing

    for (int i = 3; i >= 0; i--)
      System.out.println(i);
    System.out.println("Blastoff!!");

    int count = 0;
    for (int i = 1; i < 50; i+= 2) {
      System.out.print(i + ", ");
      count++;
    }
    System.out.printf("\nNumber of odd numbers < 50: %d\n", count);

    for (char c = 'a'; c <= 'z'; c++)
      System.out.print(c + "*");
    System.out.println(); //used for spacing
  }
}
```

The output from the above code:

```
0 1 2 3 4 5 6 7 8 9
H:e:l:l:o:!: :H:o:w: :a:r:e: :y:o:u:?:
3
2
```

```
1
0
Blastoff!!
0, 2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36, 38, 40, 42, 44, 46, 48,
Number of odd numbers < 50: 25
a*b*c*d*e*f*g*h*i*j*k*l*m*n*o*p*q*r*s*t*u*v*w*x*y*z*
```

5.2 while loops

while loops are generally used when you don't know when you are going to end. For example, if you are waiting for a change in the system or for an action from the user (versus knowing you'll end after the 100th run every time like a for loop).

General syntax:

```
initialize
while (termination condition) {
  // code
  update
}
```

A few examples:

```
public class WhileLoops {
  public static void main (String [] args) {
    int i0 = 0;
    while (i0 < 10) {
      System.out.print(i0 + " ");
      i0++;
    }
    System.out.println(); // used for spacing

    String s = "Hello! How are you?";
    int i1 = 0;
    while (i1 < s.length()) {
      System.out.print(s.charAt(i1) + ":");
      i1++;
    }
    System.out.println(); // used for spacing

    int i2 = 1, count = 0;
    while (i2 < 50) {
      System.out.print(i2 + ", ");
      i2 += 2;
      count++;
    }
    System.out.printf("\nNumber of odd numbers < 50: %d\n", count);
  }
}
```

The output from the above code:

```
0 1 2 3 4 5 6 7 8 9
H:e:l:l:o:!: :H:o:w: :a:r:e: :y:o:u:?:
0, 2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36, 38, 40, 42, 44, 46, 48,
Number of odd numbers < 50: 25
```

5.3 do-while loops

do-while loops are very similar to while loops but do-while loops always execute the code inside the brackets **at least once**.

General syntax:

```
initialize
do {
  // code
  update
} while (termination condition);
```

A few examples:

```
import java.util.Scanner;

public class DoWhileLoops {
  public static void main (String [] args) {
    int i0 = 0;
    do {
      System.out.print(i0 + " ");
      i0++;
    } while (i0 < 10);
    System.out.println(); // used for spacing

    String s = "Hello! How are you?";
    int i1 = 0;
  do {
      System.out.print(s.charAt(i1) + ":");
       i1++;
    } while (i1 < s.length());
    System.out.println();

    int i2 = 1, count = 0;
    do {
      System.out.print(i2 + ", " );
      i2 += 2;
      count++;
    } while (i2 < 50);
    System.out.printf("\nNumber of odd numbers < 50: %d\n", count);

    Scanner reader = new Scanner (System.in);
    String response = "";
    do {
      System.out.print("Are we there yet? ");
      response = reader.nextLine();
    } while (!response.equalsIgnoreCase("yes"));
    System.out.println("Finally!!!");
    reader.close();
  }
}
```

The output from the above code:

```
0 1 2 3 4 5 6 7 8 9
H:e:l:l:o:!: :H:o:w: :a:r:e: :y:o:u:?:
0, 2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36, 38, 40, 42, 44, 46, 48,
Number of odd numbers < 50: 25
Are we there yet? no
Are we there yet? almost...
Are we there yet? still not there
Are we there yet? yes
Finally!!!
```

5.4 Suggestions, Warnings, and Resources

- Resource: For loops resource
- Resource: While loops resource
- Resource: Do-While loops resource
- Resource: Java Documentation for break and continue statements

- OPTIONAL: For-Each Loop Tutorial
- Warning: Watch the ranges! When a String's length is of size 8, i $\leq$ 9 is the same as i $\leq$ 8. However, i $\leq$ 9 will give you an error.
- Remember: You can always change your update by using the "+=" or "-=" method, especially if you are updating by more than 1.
- Remember: 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!
- 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.

5.5 FAQs

1. **Q:** When do you use one loop over another?
   A: You can almost always use any loop you choose, but there maybe one that is easier than the other. For example, in the "Are we there yet?" example from above, using a do-while is nice because the loop always runs at least once (you could have also used a while loop but you would have had to ask the "Are we there yet?" question before and in the while loop).

6 Classes

General Syntax: `ClassName.MethodName(parameters if applicable);`

6.1 Math

Common Math Methods and Constants:

- sqrt (ex. `Math.sqrt(64);`)
- pow (ex. `Math.pow(1.0, 3.0);`)
- sin (ex. `Math.sin(1.5);`)
- cos (ex. `Math.cos(1.5);`)
- exp (ex. `Math.exp(3.3)`)
- log (ex. `Math.log(2.1);`)
- min (ex. `Math.min(1, 3);`)
- max (ex. `Math.max(1, 3);`)
- round (ex. `Math.round(3.3)`)
- floor (ex. `Math.floor(3.3);`)
- ceil (ex. `Math.ceil(3.3)`))
- PI (constant) (ex. `double num = Math.PI;`)

6.2 Character

Common Character Methods

- isUpperCase (ex. `Character.isUpperCase('a');`)
- isLowerCase (ex. `Character.isLowerCase('a');`)
- isLetter (ex. `Character.isLetter('4');`)
- isDigit (ex. `Character.isDigit('4');`)

6.3 Examples

```
import java.util.Scanner;

public class OtherClasses{
  public static void main (String [] args) {
    String s = "My ID is 12345";
    int upper = 0, lower = 0, number = 0;
    for (int i = 0; i < s.length(); i++) {
      if (Character.isUpperCase(s.charAt(i)))
        upper++;
      else if (Character.isLowerCase(s.charAt(i)))
        lower++;
      else if (Character.isDigit(s.charAt(i)))
        number++;
    }
    System.out.printf("Number of Upper: %d\nNumber of Lower: %d\nNumber of Digits: %d\n",
                      upper, lower, number);

    Scanner reader = new Scanner (System.in);
    System.out.print("Enter an angle (in radians): ");
    double angle = reader.nextDouble();
    System.out.printf("Equivalent to your angle is: %.3f\n", angle + 2* Math.PI);
    System.out.printf("sin(%f) = %.3f\n", angle, Math.sin(angle));
    System.out.printf("cos(%f) = %.3f\n", angle, Math.cos(angle));
    reader.close();
  }
}
```

Output from the code above:

```
Number of Upper: 3
Number of Lower: 3
Number of Digits: 5
Enter an angle (in radians): 1.34289823981
Equivalent to your angle is: 7.626
sin(1.342898) = 0.974
cos(1.342898) = 0.226
```

7 Practice Written Exam

7.1 Short Answer

1. Print the pre-defined double variable `d` with 3 decimal places (with a new line).
2. Declare a Scanner called `keys`.
3. Read one integer from `keys` and store into the pre-defined int variable `num1`.
4. Read one double from `keys` and store into the pre-defined double variable `num2`.
5. Print the variables from question 2 and 3 in the following format: `num1 num2`
6. Read in one word and store it into the predefined String variable called `word`.
7. Print out the second through fourth character of `word`.
   Note: You can assume the word is at least 4 characters long.
8. Read in one line and store it into the predefined String variable called `line`.
9. Close `keys`.
10. Given the following code, write what would be printed to the console (write error if the code throws an error):

```
String s = new String("Bob Marley");
String m = "mummies";
System.out.println(m.substring(0,3) + s.substring(4));
System.out.print(m.charAt(1) + "-" + s.charAt(1));
System.out.println(s.indexOf('q') + m.indexOf(3));
System.out.println(s.length());
System.out.println(s + m);
System.out.println(m.indexOf("Ma"));
System.out.println(s.length() - m.length());
System.out.println(s.substring(0,11));
System.out.println(s.length()/m.length());
```

11. Write an if/else if/else statement for the following information (`year` is already declared):
    - if year is 1970, print "history is cool"
    - if year is 1980, print "Era of Hippies!"
    - if year is 1990, print "Cassette Tapes!"
    - if year is 2000, print "iPhone's begin their take over..."
    - if year is 2010, print "US wished Queen Elizabeth II happy birthday on the wrong day, good start to the decade..."
    - if year is anything else, print "Huh, I'm not sure what to say"
12. Write a switch statement based off the same information from the previous question
13. Given the following code, what is printed?
    `String s = "Roses are red, violets are blue,...";`
    (a) `System.out.println(s.substring(0));`
    (b) `System.out.println(s.substring(6, 8));`
14. Write a for loop that prints the numbers 3 to 8 separated by a comma. It is okay to have a trailing comma at the end.
15. Write a while loop that prints the numbers 3 to 8 separated by a comma. It is okay to have a trailing comma at the end.

16. Write a do-while loop that prints the numbers 3 to 8 separated by a comma.It is okay to have a trailing comma at the end.

17. Using a loop (of any kind) print all numbers that is a multiple of three and that is between 1 and 50 (inclusive) separated by semicolons.

18. Using a loop (of any kind) print each character of the pre-defined String `s` separated by a newline.

19. Using a loop (of any kind) print the pre-defined String `s` backwards (characters all on the same line).

20. Using a loop (of any kind) print every other letter of the pre-defined String variable `s` (characters all on the same line).

7.2 Tracing

For each example, write what is printed. For questions with user input look at the input section below.

```
import java.util.Scanner;

public class Tracing {
  public static void main (String [] args) {
    String s = "abcdefg";
    // Question 1
    for (int i = 0; i < s.length() - 1; i++)
      System.out.print(s.charAt(i) + ".");
    System.out.println(); // used for spacing

    boolean b = true;
    int num = 3;
    while (b) {
      ++num;
      if (num < 5)
        continue;
      else
        b = false;
    }
    // Question 2
    System.out.println(num);

    Scanner scan = new Scanner (System.in);
    String input = "";
    int count = 0;
    System.out.println("Enter a word. If you enter 'stop' the loop ends");
    do {
      count++;
      input = scan.next();
    } while (!input.equals("stop"));
    // Question 3
    // look at the console run below
    System.out.println("The loop ran " + count + " times.");

    System.out.println("Enter two lines of text: ");
    String word = scan.next();
    String first_line = scan.nextLine();
    String second_line = scan.nextLine();
    // Question 4
    System.out.println("first word: " + word);
    // Question 5
    System.out.println("first line: " + first_line);
    // Question 6
    System.out.println("second line: " + second_line);
    String lines = first_line + second_line;
    int upper = 0, lower = 0, number = 0, character = 0;
    for (int i = 0; i < first_line.length() + second_line.length(); i++) {
      if (Character.isUpperCase(lines.charAt(i)))
        upper++;
      if (Character.isLowerCase(lines.charAt(i)))
        lower++;
      if (Character.isLetter(lines.charAt(i)))
        character++;
      if (Character.isDigit(lines.charAt(i)))
        number++;
    }
    // Question 7
    System.out.println("Number of upper case letters: " + upper);
    // Question 8
    System.out.println("Number of lower case letters: " + lower);
    // Question 9
    System.out.println("Number of letters: " + character);
    // Question 10
    System.out.println("Number of digits: " + number);

    scan.close();
```

```
  }
}
```

User input:

```
// input for question 3
Enter a word. If you enter 'stop' the loop ends
stop

// input for questions 4-10
Enter two lines of text:
hey! My id number Is
42319!
```

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 in the program, you change your variables 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

Redo past recitations and assignments until you no longer need to use the internet, friends, or past code.

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
- Semicolons right after loops and if statements

8.4 Challenges

CodingBat and Hackerrank offer good extra coding practice.

ANSWERS ON THE NEXT PAGE

9 Answers to Practice Written Problems

9.1 Short Answer

1.
```
    System.out.printf("%.3f\n", d);
```

2.
```
Scanner keys = new Scanner (System.in);
```

3.
```
num1 = keys.nextInt();
```

4.
```
num2 = keys.nextDouble();
```

5.
```
System.out.println(num1 + " " + num2);
```

6.
```
word = keys.next();
```

7.
```
System.out.println(word.substring(1,4));
```

8.
```
line = keys.nextLine();
```

9.
```
keys.close();
```

10.
```
mumMarley
u-o-2
10
Bob Marleymummies
-1
3
error
1
```

11.
```
if (year == 1970)
    System.out.println("history is cool");
else if (year == 1980)
    System.out.println("Era of Hippies!");
else if (year == 1990)
    System.out.println("Cassette Tapes!");
else if (year == 2000)
    System.out.println("iPhone's begin their take over...");
else if (year == 2010)
    System.out.println("US wished Queen Elizabeth II happy birthday on the wrong day, good
     start to the decade...");
else
    System.out.println("Huh, I'm not sure what to say");
```

12.
```
switch (year){
case 1970: System.out.println("history is cool"); break;
case 1980: System.out.println ("Era of Hippies!"); break;
case 1990: System.out.println("Cassette Tapes!"); break;
case 2000: System.out.println ("iPhone's begin their take over..."); break;
case 2010: System.out.println ("US wished Queen Elizabeth II happy birthday on the wrong
    day, good start to the decade..."); break;
default: System.out.println ("Huh, I'm not sure what to say"); break;
```

13. (a) Roses are red, violets are blue,...

    (b) ar

14.
```
// could also have (i < 9)
for (int i = 3; i <= 8; i++)
  System.out.print(i + ",");
```

15.
```
// could also have (i1 <= 8)
// could also increment outside of print
int i1 = 3;
while (i1 < 9)
  System.out.print(i1++ + ",");
```

16.
```
// could use pre or post increment or increment like in the previous question
int i2 = 3;
do {
  System.out.print(i2 + ",");
  i2++;
} while (i2 <= 8);
```

17.
```
for (int i = 0; i <= 50; i++)
  if (i % 3 == 0)
     System.out.print(i + ";");
```

18.
```
for (int i = 0; i < s.length(); i++)
  System.out.println(s.charAt(i));
```

19.
```
for (int i = s.length() - 1; i >= 0; i--)
  System.out.print(s.charAt(i));
```

20.
```
for (int i = 0; i < s.length(); i+=2)
  System.out.print(s.charAt(i));
```

Note: For answers 17 - 20, your implementation may be different (¡ vs ¡=, while, do-while, or for, etc). These answers are just for guidance and there are many ways to correctly implement these questions.

9.2 Tracing

1. `a.b.c.d.e.f.`
2. `5`
3. `first word:  hey!`
4. `first line:  My id number Is`
5. `second line:  42319!`
6. `Number of upper case letters:  2`
7. `Number of lower case letters:  10`
8. `Number of letters:  12`
9. `Number of digits:  5`