

CS 163/164 – Fall 16  
Exam 1 Prep



Declare and assign a float variable called **f** to 9.

```
float f = 9F;
```

Initialize a Scanner called **reader** that reads from the keyboard.

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

Name the five errors in the following code:

```
public static void main (String [] args){  
    String s = "Hello";  
    int i = s.charAt('e');  
    for (int i = 0; i <= s.length(); i++)  
        if (s.charAt(i) == 'a');  
        System.out.println("cool it's an a");  
    if (s == "Hello")  
        s+= " World";  
}
```

1. charAt takes a number and returns a character, what I should have put was s.indexOf(some int)
2. int i is redeclared within the for loop.
3. For loop goes one too far. Should have put i < s.length() or i < s.length() - 1.
4. semicolon after the if statement inside the for loop
5. must check equality of Strings with .equals NOT ==. So instead it should be s.equals("Hello");

Name the 8 primitive types

char, int, double, float, long,  
short, byte, boolean

Read in a double from the Scanner **reader**  
and store it into the pre-defined variable **d**.

```
d = reader.nextDouble();
```

What is the difference between declaring, assigning, and initializing?

Declaring just tells the computer that you'll use this variable sometime. Example: `char c;`

Assigning just assigns a value to the variable. Example `c = 'a';`

Initializing does declaring and assigning in one step. Example `int i = 3;`

**Note:** You only declare or initialize once per variable, but you can assign (or re-assign) as many times as you want

What do the following lines print?

```
System.out.printf("%.2f\n", 3.1415);  
System.out.printf("%.3f\n", 3.1415);  
System.out.printf("%f\n", 3.1);  
System.out.printf("%.4f\n", 3.1);  
System.out.printf("%.5f\n", (double)3);  
System.out.printf("%.5f\n", 3);
```

```
3.14  
3.142  
3.100000  
3.1000  
3.00000  
Illegal Formatting Exception
```

Write a for loop that prints each character in the predefined variable String **s** separated by an ampersand (&) all on the same line.



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

Close the Scanner called **reader**.

```
reader.close();
```

Store the result when you check if the predefined String variables **s1** and **s2** are equal into the pre-defined variable **b**.

```
b = s1.equals(s2);
```

Store the result when you check if the predefined int variables **i1** and **i2** are equal into the pre-defined variable **b**.

```
b = i1 == i2;
```

Print the following predefined variables using **printf**, all on the same line, separated by ampersands (&), ending with a new line.

double d (with 6 decimal accuracy)

char c

string s

double d1 (with 2 decimal accuracy)

```
System.out.printf("%f&%c&%s&%.2f\n", d, c, s, d1);
```

Based on the predefined String variable s.  
Write an if-else if-else statement:

- when the length of s is less than 3 print "short word"
- when the length of s is more than 12 print "long word"
- for any other word print "between 3 and 12"

```
if (s.length() < 3)
    System.out.println("short word");
else if (s.length() > 12)
    System.out.println("long word");
else
    System.out.println("between 3 and 12");
```

Use the predefined Scanner **keys** to read and store the *first* word into the predefined String variable **s**.

Line:  
Hello World! How's it going?

```
s = keys.next();
```

Use the predefined Scanner **keys** to read and store the following *line* into the predefined String variable **s**.

Line being read:  
Hello World! How's it going?

```
s = keys.nextLine();
```

Declare a Scanner called **input** that reads the following information (and in this order) and stores into respective variables that you must declare:

- word
- double
- first character of the next word
- the rest of the line
- an integer
- the full next line



```
Scanner input = new Scanner (System.in);
String word, line0, line1;
double d;
char c;
int i;

word = input.next();
d = input.nextDouble();
c = input.next().charAt(0);
line0 = input.nextLine();
i = input.nextInt();
input.nextLine();
line1 = input.nextLine();
```

Write a switch statement based off of the String variable **str**. If **str** equals "Bob" print "Marley", if **str** equals "Michael" print "Jackson", if **str** equals "Justin" print "Timberlake", if **str** is none of those options print "okay then".

\*\* NOTE: "miChaeL" and "MICHAEL" should still print "Jackson".

```
switch (str.toLowerCase()){  
  case "bob": System.out.println("Marley"); break;  
  case "michael": System.out.println("Jackson"); break;  
  case "justin": System.out.println("Timberlake"); break;  
  default: System.out.println("okay then");  
}
```

What does the following code print?

```
char c = '@';  
switch (c) {  
  case '@': System.out.println("char c = '@");  
  case '$': System.out.println("char c = '$");  
  case '4': System.out.println("char c = '4");  
  case 'u': System.out.println("char c = 'u");  
  case ' ': System.out.println("char c = ' ");  
  default: System.out.println("char c is not '@', '$', '4', 'u', ' ');  
}
```

```
char c = '@'  
char c = '$'  
char c = '4'  
char c = 'u'  
char c = ''  
char c is not '@', '$', '4', 'u', ''
```

What does the following code print?

```
String s = "Cool bro";  
System.out.println(s.substring(2, 6));
```

ol b

What does the console print based off the following code?

```
public class Practice {  
    public static void main (String [] args){  
        String s = "Koala Bears";  
        for (int i = 0; i < s.length(); i+=2)  
            System.out.print(s.charAt(i));  
    }  
}
```

KaaBas

Write a while loop that prints the numbers 3  
– 9 (inclusive) all on new lines.

```
int i = 3;
while (i <= 9) {
    System.out.println(i);
    i++;
}
```

Write an if statement that prints “between 0 and 100” on a new line, when the predefined int variable `i` is between 0 and 100 (inclusive).

```
if (i >= 0 && i <= 100)
    System.out.println("between 0 and 100");
```

Write an if statement that prints "char is an a or a c" on a new line, when the predefined variable char **c** is either 'a' or 'c'.

```
if (c == 'a' || c == 'c')  
    System.out.println("char is an a or a c");
```

What does the following code print?

```
for (int l = 0; l > 0; l++)  
    System.out.println(l);
```



nothing

Which one of these would not go to the end of String s? (Specify which ones cause errors and which ones are just incorrect (but they compile)?

- A. `for (int i = 0; i <= s.length(); i++)`
- B. `for (int i = 0; i < s.length()+1; i++)`
- C. `for (int i = 0; i < s.length(); i++)`
- D. `for (int i = 0; i <= s.length() -1; i++)`
- E. `for (int i = 0; i < s.length() -1; i++)`

A gives an error (goes one more)  
B gives an error (goes one more)  
C is correct  
D is correct  
E is incorrect only goes to the second to the last index

Write a do-while loop that will add the sum of all numbers from 1 - 19 (inclusive).

```
int count = 1;
int sum = 0;
do {
    sum += count;
    count++;
} while(count < 20);
```

Find the 4 errors within the block of code

```
public static void main (String [] args){
    long absurdlyLong = 234897523456476345;
    System.out.printf("%s%n", hola);
    System.out.println(absurdlyLong);
    System.out.println(double(3));
}

public static int double(int num) {
    for(int i = 0; i <= num; i++) {
        int counter = 0;
        counter *= num;
    }
    return counter;
}
```

1.No L following the long value.

2.double is a reserved keyword, I would not be able to name my method double.

3.Counter is declared within the for loop, not a valid return variable.

4.hola has never been declared, if I wanted to print hola, I would need to place quotes around the word, or declare it as a variable somewhere else.