

Introduction to Java

In this course, you will work in teams of 3–4 students to learn new concepts. This activity will introduce you to the process. We'll also take a first look at how to store data in Java programs.

Content Learning Objectives

After completing this activity, students should be able to:

- Describe the responsibility of each team role.
- Identify components of the "hello world" program.
- Write Java code to declare int and double variables.
- Explain what it means to assign a value to a variable.

Process Skill Goals

During the activity, students should make progress toward:

- Leveraging prior knowledge and experience of other students. (Teamwork)

Facilitation Notes

This activity is designed for the first day or week of CS1. There are no prerequisites. During this activity, it's important to explain/model the instructor's role in an active learning classroom. For large classes, it may take a bit longer than expected to organize teams.

The meta activity refers to [Role Cards](#), which ideally should be printed out on a different color of card stock (one set per team). You might want to have each team complete a [Team Report](#) at the end of this activity.

One strategy for reporting out **Model 1** is to have presenters of nearby teams swap answers and discuss any differences. Introduce the terms *documentation* (see #7) and *newline* (see #8).

Questions that ask students to explain (or to describe) are good for reporting out. It's important that teams hear multiple perspectives on these concepts. Introduce the terms *data type* and *operator*, and point out that *assignment* is not the same as *equality*.

On **Model 3**, it may be necessary to explain that there are only two variables. The boxes show the state of memory over time, as the corresponding statements are executed.

Key questions: #7, #13, #20

Source files: [HelloWorld.java](#)



Copyright © 2021 Chris Mayfield and Helen Hu. This work is licensed under a Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License.

Meta Activity: Team Roles

Decide who will be what role for today; we will rotate the roles each week. If you have only three people, one should have two roles. If you have five people, two may share the same role.

Manager:	Helen Hu
Presenter:	Clif Kussmaul
Recorder:	Chris Mayfield
Reflector:	Aman Yadav

keeps track of time, all voices are heard

asks questions, gives the team's answers

quality control and consensus building

team dynamics, suggest improvements

Questions (15 min)

Start time:

1. What is the difference between **bold** and *italics* on the role cards?

The bold points describe what the responsibilities are. Examples of what that person would say are in italics.

2. Manager: invite each person to explain their role to the team. Recorder: take notes of the discussion by writing down key phrases next to the table above.

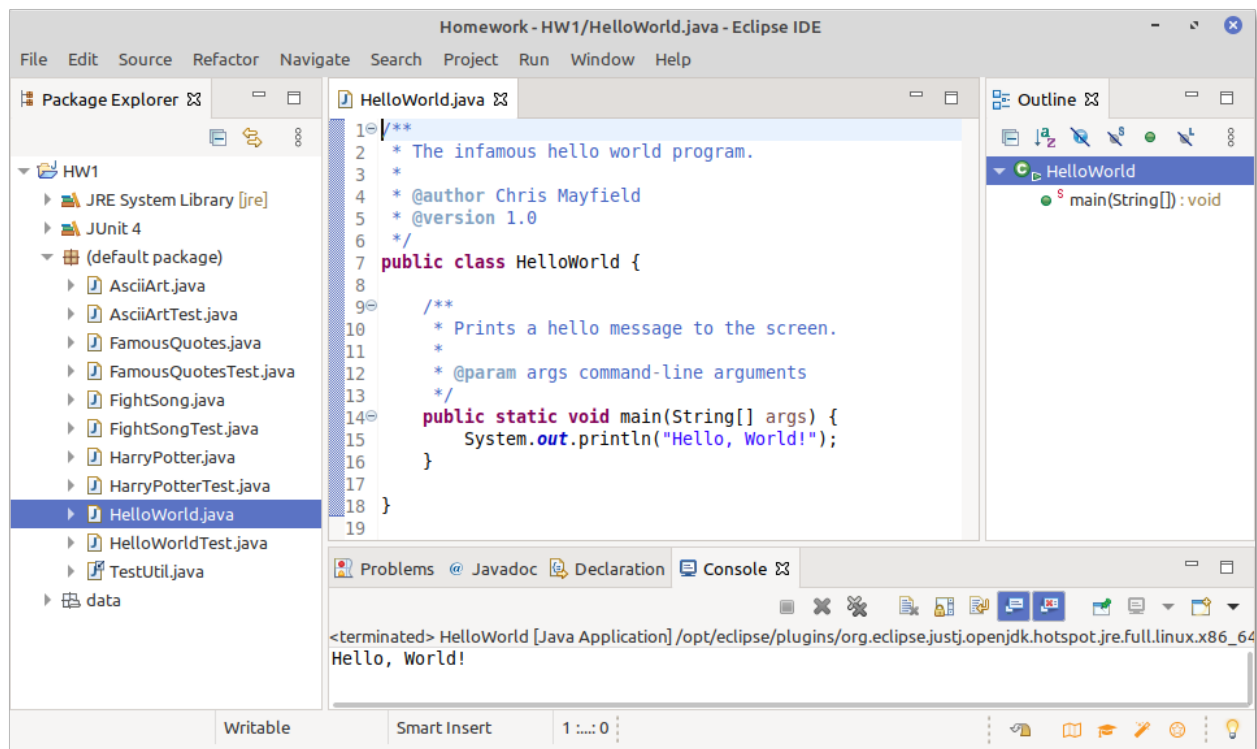
3. What responsibilities do two or more roles have in common?

Both the presenter and the recorder help the team reach consensus. The manager and reflector both monitor how the team is working.

4. For each role, give an example of how someone observing your team would know that a person is not doing their job well.

- Manager: The team is constantly getting behind.
- Presenter: The student doesn't know what to say.
- Recorder: Some team members aren't taking good notes.
- Reflector: The student never comments on team dynamics.

Model 1 Hello, World!



Questions (10 min)

Start time:

5. What is the name of the class? What is the name of the file? What folder is the file in?

Class name: HelloWorld File name: HelloWorld.java Directory: HW1/

6. How many lines of code is the above program? How many statements does it have?

The source file has 18 lines. There is only one statement (the `System.out.println`).

7. What is the purpose of the first six lines? What is the purpose of the two blank lines?

The first six lines describe what the program does and who wrote it.
The two blank lines make it easier to read the program (they are ignored).

8. Describe in your own words what `System.out.println` does. Be very specific.

The `println` method displays a message on the screen, followed by a newline character. When you run the HelloWorld program, it prints a welcome message.

Model 2 Variables

Most programs store and manipulate data values. We use *variables* to give values meaningful names. The following code *declares* three variables and *assigns* them (using the = operator). Each variable is stored in the computer's memory, represented by the boxes on the right.

Java code

```
int dollars;  
int cents;  
double grams;  
  
dollars = 1;  
cents = 90;  
grams = 3.5;
```

Computer memory

dollars	1
cents	90
grams	3.5

Questions (10 min)

Start time:

9. Identify the Java *keyword* used in a variable declaration to indicate

a) an integer:

b) a real number:

10. Consider numbers of dollar bills, cents, and grams. Which of these units only makes sense as an integer, and why?

Cents makes sense (ha ha) only as an integer, because at the end of the day, you can't pay with a fractional amount. (It is possible to make a similar argument for dollars, but not for grams.)

11. What would you expect the following statements to print out?

a) `System.out.println(dollars);`

b) `System.out.println(cents);`

c) `System.out.println(grams);`

12. In the previous question, how does the third printed line (c) differ from the first two?

The third line prints a double, and the first two print an integer.

13. What do you think is the purpose of a variable declaration?

It tells the computer how to interpret and display the value.

14. What will be output by the following code, and why?

```
double one;  
one = 1;  
System.out.println(one);
```

The output is 1.0, because one is a double. The type of variable determines the output format.

Model 3 Assignment

The following eight lines of code are executed one at a time, in order. The boxes on the right show the state of the (same) two variables after each step.

Declaring a variable instructs the computer to reserve space for it in memory:

```
1 int dollars;  
2 int cents;
```

dollars

cents

Variables cannot be used until they are *initialized* (assigned for the first time):

```
3 dollars = 2;  
4 System.out.println(dollars); // OK  
5 System.out.println(cents); // error
```

dollars

cents

Each time you assign a variable, you are *updating* its value stored in memory:

```
6 dollars = 3;  
7 dollars = 4;  
8 cents = 49;
```

dollars

cents

Questions (10 min)

Start time:

15. How many times is each variable in Model 3 assigned?

The variable dollars is assigned three times, but cents is assigned only once.

16. What is the error in the second `System.out.println` statement? (Don't just repeat the text in Model 3; explain in your own words what the problem is.)

The variable `cents` is not initialized, so Java does not know what value to print.

17. What is the value of `dollars` right before it's assigned for the last time? What is the value of `cents` before it's assigned for the last time?

Just before the `dollars = 4;` statement, `dollars` is 3. And before the `cents = 49;` statement, `cents` is uninitialized.

18. Consider the statement: `cents = dollars;`

a) Compare this code to lines 6–8 in Model 3. What value do you think `cents` and `dollars` will have after running this statement?

The variable `cents` will be 4, and `dollars` will remain unchanged.

b) Which side of the equals sign (left or right) was assigned a new value?

The left side.

19. In Java, the `+` and `-` symbols are used to perform addition and subtraction. For example, the statement `dollars = dollars + 1;` adds one to the current value of `dollars`.

a) What is the value of `dollars` (in memory) after running this statement? 5

b) Do you consider the equals sign in Java an operation to be performed? (like `+`) If so, explain the operation. If not, explain why not.

Yes; it executes the assignment operation which stores a value in memory.

c) Do you consider the equals sign in mathematics an operation to be performed? If so, explain the operation. If not, explain why not.

No; it simply states the proposition that two values are equal.

20. In your own words, explain how you should read the `=` sign in Java. For example, the Java statement `x = a + b;` should be read out loud as "x _____ a plus b."

Answers may include "x gets a plus b", "x becomes a plus b", etc.