

Solve the following problems. If necessary, include code, images, or scanned drawings to support your answer. Submit your written solutions as a pdf. Attach Java source files as needed. Submit all files to Canvas.

Question:	Total
Points:	0
Score:	



Week 1 Lab


1 Objectives


1. Learn how to use IntelliJ.
2. Practice using Java commands.
3. Practice working with literals and expressions.
4. Practice working with variables.
5. Practice working with mathematical functions.
6. Practice manipulating strings.
7. Learn how to submit an assignment to Canvas.

2 Lab Teams

Your lab instructor will assign you into groups of two. You will work with this group for the remainder of the semester.

Each week the partner in the face-to-face lab will be the Driver . The online partner is the Navigator .

 The Driver enters the code and explains what the code does as it is being typed-in. The Driver provides suggestions for the Navigator who answers the written questions.

 The Navigator answers the written questions and offers suggestions, testing values, and determines when the lab problem is done and ready to submit.

You must swap roles every week. By the end of the course, you must have performed each role at least three times. We will provide prompts indicating each individual tasks in the lab assignments.

Remember, coding is a social activity. Sharing of ideas and code is encouraged among teammates within the lab. Pull the best code ideas together for your final submission.

2.1 Questions for Reflection

1. Over the course of the semester, you will develop a Lab Portfolio containing all of the team's notes, discussions, and problem solutions.

Create a Google shared drive (<https://drive.google.com>). Share the drive with all team mates, your lab instructor.

2. Record the following information about each of your lab partners: **Name, MTU eMail, Phone, Major, Common study time outside class or lab.**

3. Negotiate the responsibilities for each role as you work through the lab. Figure out what works best for you and report on this in your Lab Portfolio. Think about how you will cooperate as a team. Ask questions, share information, take notes.

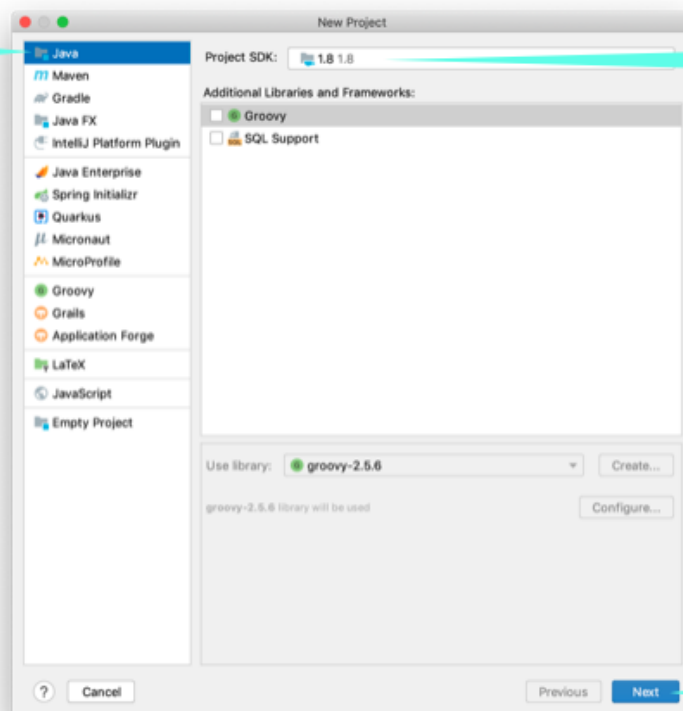
3 Background

3.1 Creating a new project and Running Java



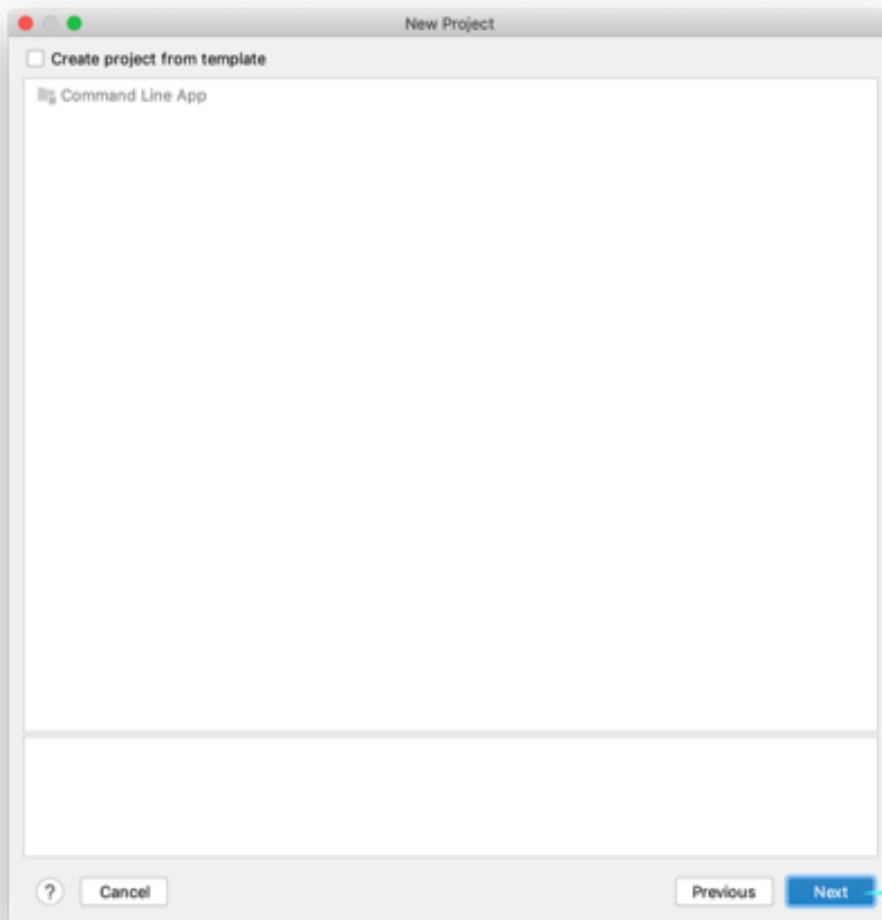
Create a new Project

1. Select Java



2. Select JDK 1.8

3. Click Next



Click Next

New Project

Project name:

Project location:

More Settings

Module name:

Content root:

Module file location:

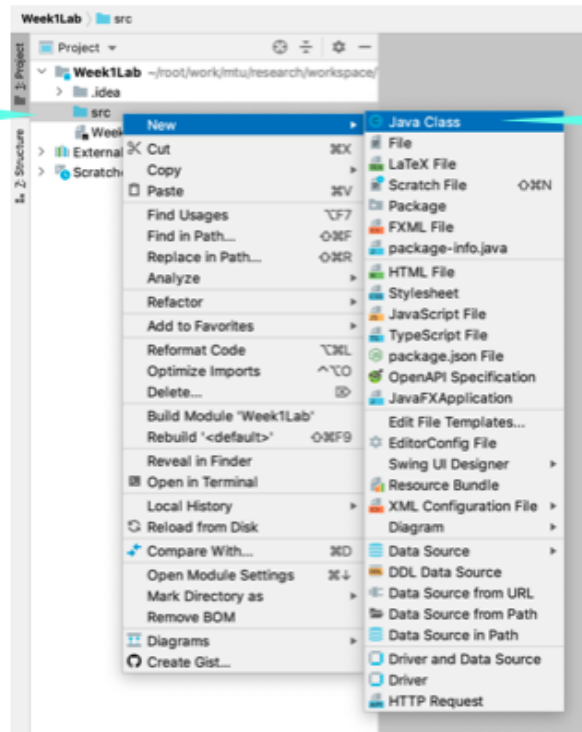
Project format:

? Cancel Previous Finish

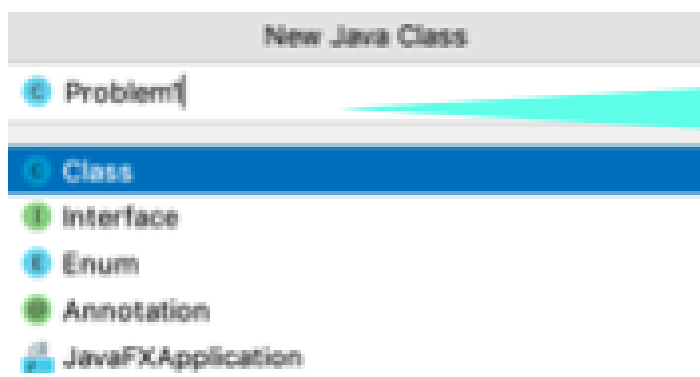
1. Enter the project name

2. Click Finish

1. Right-click on the src folder

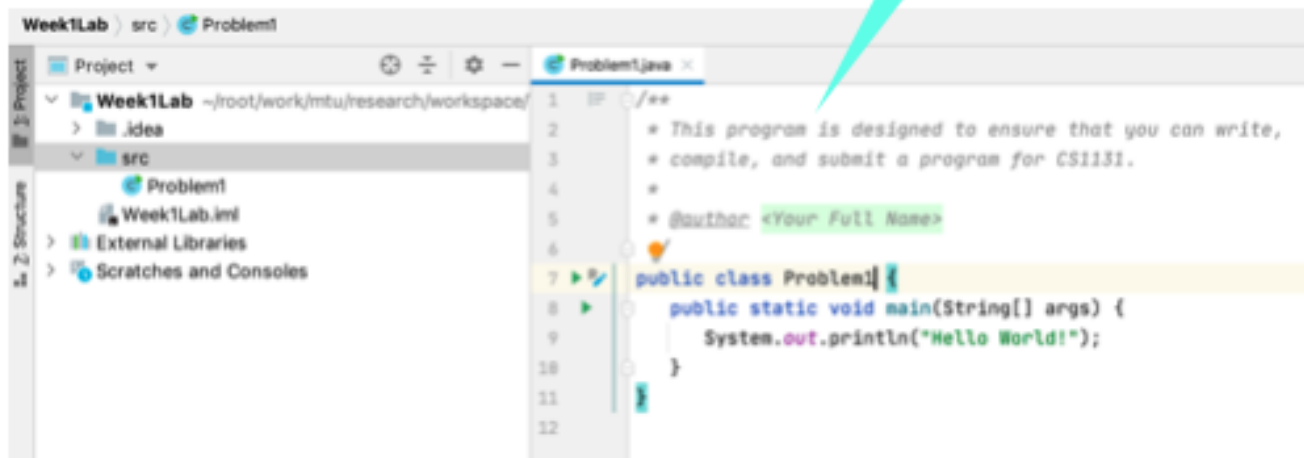


2. Choose new, then Java Class



Enter the class name

Enter Java code
here.

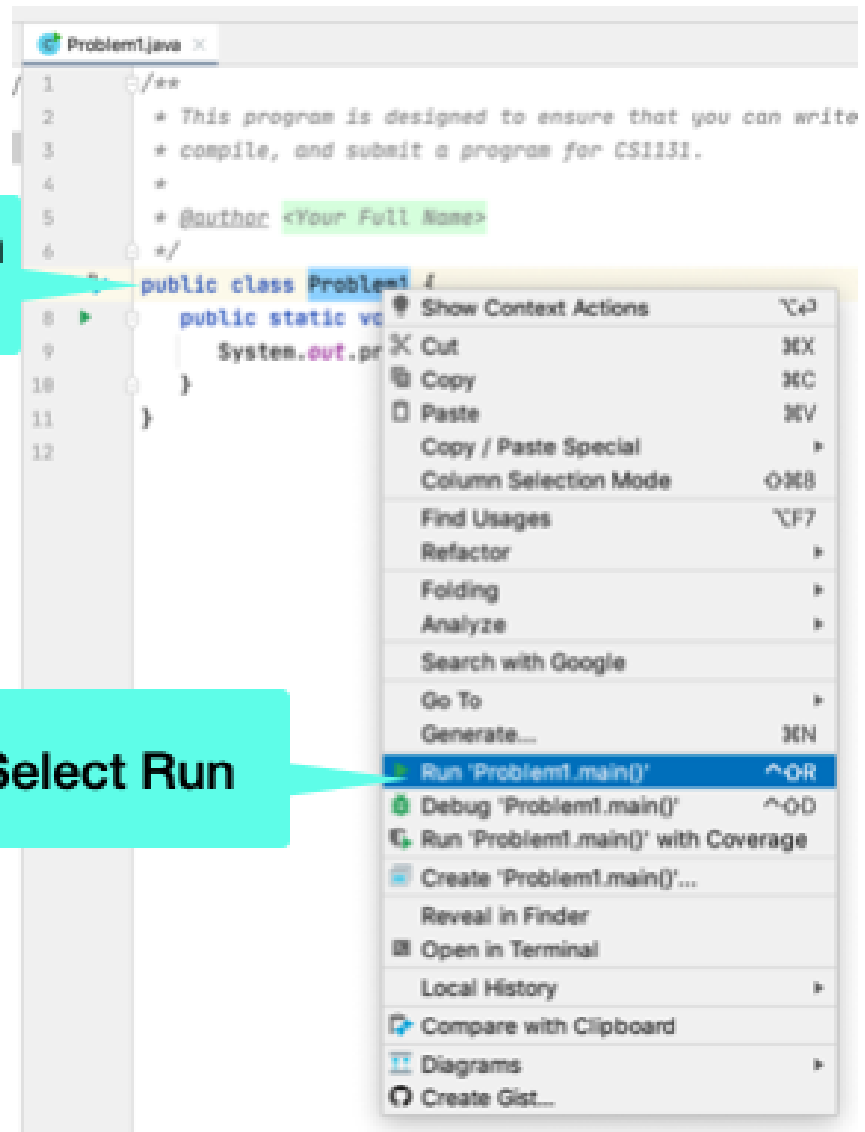


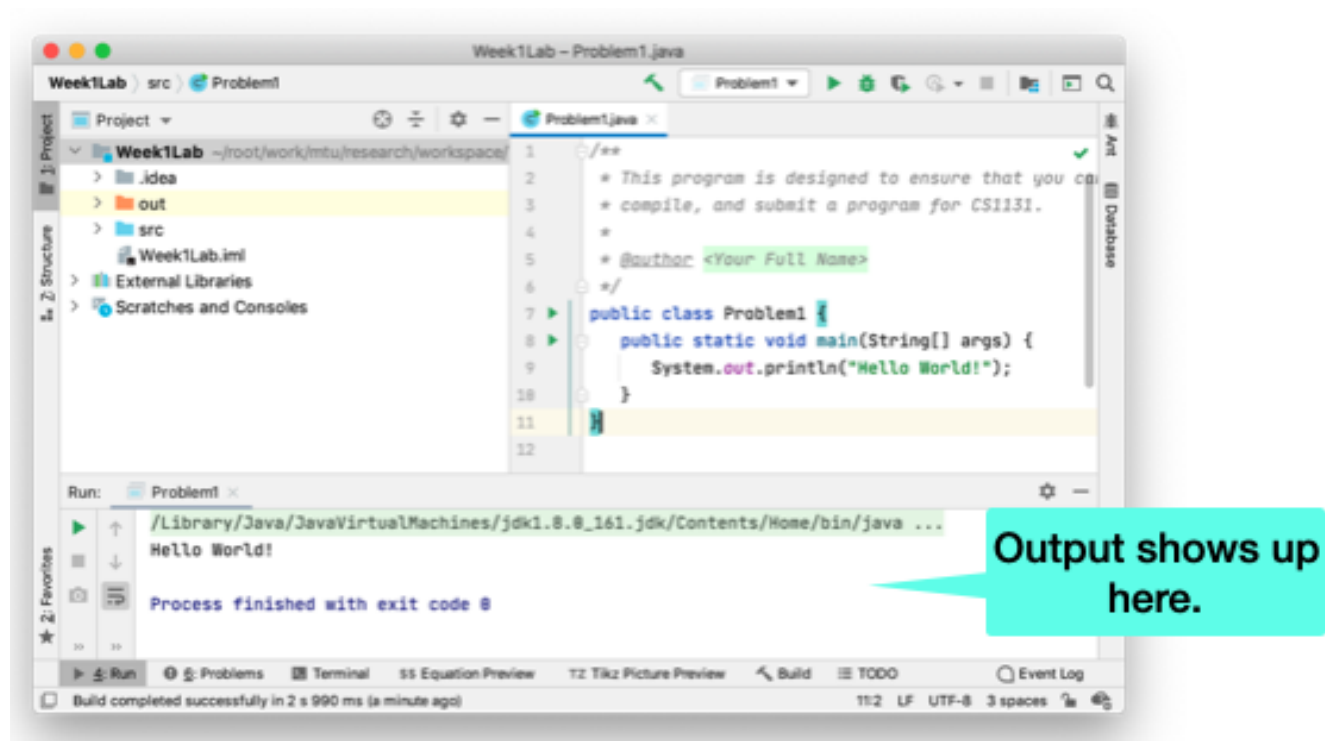
The screenshot shows an IDE window titled "Week1Lab" with a project named "Problem1". The left sidebar shows the project structure with "src" containing "Problem1". The main editor displays the following Java code:

```
1  /**
2   * This program is designed to ensure that you can write,
3   * compile, and submit a program for CS1131.
4   *
5   * @author <Your Full Name>
6   */
7  public class Problem1 {
8      public static void main(String[] args) {
9          System.out.println("Hello World!");
10     }
11 }
12
```

1. Right-click on the class name

2. Select Run





3.2 Packages

Java is a library-based language. Most of the commands used in Java programming are code written by programmers such as yourself. These commands are bundled together in packages that form libraries of routines that can be used by anyone. We will mostly use the core java commands, in the `java.lang` package. Sometimes we will use other packages, such as the `java.util` package, which contains a collection of useful utilities. Anytime we want to use a command that is not in the core language, we have to tell Java which package to use. We do this by placing an `import` command at the top of our source code file.

3.3 User Input

In order to read input, you will be using `System.in` and `Scanner`. To use the `Scanner` we have to import its library. The following code imports the `Scanner` and uses it to obtain data from the user. Enter the code and run it to get an idea of how this works.

You can read in an `int` using `input.nextInt()`. You can read in a `double` using `input.nextDouble()`. There are many other types which `Scanner` supports, but for the purposes of this lab you will only need the previous two method calls. If you are interested in the other functionalities provided please consult the Javadocs. <https://docs.oracle.com/javase/8/docs/api/index.html>

3.4 Try running the following code in IntelliJ

```
1 import java.util.Scanner;
2 public class MilesPerGallon {
3     public static void main( String [ ] args ) {
4         Scanner input = new Scanner( System.in );
5         System.out.print( "How many miles were driven? ");
6         double miles = input.nextDouble();
7         System.out.print( "How many gallons of petrol were
            consumed? ");
8         double gallons = input.nextDouble();
9         double mpg = miles / gallons;
10        System.out.println( "Miles-per-gallon = " + mpg );
11    }
12 }
```

3.5 External Libraries

Occasionally, we will ask you to use an external library that is not include in the Java distribution. These are distributed in jar files that you will download from Canvas. A JAR (Java ARchive) is a zip file format containing many Java class files.

As an example, download the Logo.jar file from Canvas and store it in a folder called "lib" that you create under the project folder. The enter the following code:

```
1 import java.awt.Color;
2 public class Square {
3     public static void main(String[] args) {
4         Turtle t = new Turtle ( );
5
6         t.setPenWidth(5);
7         t.setPenColor(Color.BLACK);
8         t.penDown();
9         t.forward (100);
10        t.turn(90);
11        t.forward (100);
12        t.turn (90);
13        t.forward (100);
14        t.turn(90);
15        t.forward (100);
16        t.turn (90);
17    }
18 }
```

4 Completing the Lab

Before you leave the lab, show your Lab Instructor your work and ask for it to be graded and ask for feedback.

1. You will receive full credit if you complete all of the assigned tasks. 2. You will receive partial credit if you complete only some of the tasks. 3. You will receive no credit if you do not complete any of the tasks.

4.1 Group Reflection

Navigator: Create a google document called Week1LabReflection and share it with everyone on your lab team. Immediately add everyone's names to the top of the document.

Driver: Consider the group's performance today. Email one piece of positive evidence to each member of the group for one of the Effectiveness Competencies listed below. Only give each person one piece of positive feedback right now.

4.2 Score

- | | | |
|----------------------------------|-------|-----------|
| 1. Creating and running project: | _____ | / 100 pts |
| 2. MilePerGallon: | _____ | / 200 pts |
| 3. External Libraries: | _____ | / 200 pts |
| Total: | _____ | / 500 pts |

4.3 Submitting your Work

Each student must submit the work completed in lab. This provides a record of your effort in the lab.

If grade questions arise in the future, we will rely on this record to show your work.

1. Make sure you have completed and saved any critical thinking questions asked.
2. Upload all required files to Canvas:
 - Your Lab Text Document, containing: (1) lab partner details, (2) lab partner in-person calendar, (3) each individual partner(s) paper airplane instructions, (4) A summary of the instruction discussion, (5) your group's revised instructions, (6) A description of changes made in the revised set

- a. Click the **Submit Assignment** button at the top of the lab.

Week 1 Lab

Submit Assignment

Due Wednesday by 2pm Points 500 Submitting a file upload

- b. Add your files for uploading.

File Upload [Google Doc](#)

Upload a file, or choose a file you've already uploaded.

File: Hello World.xml

square.xml

First-Expression.xml

Math-Expressions.xml

Week1LabReport

[+ Add Another File](#)

[Click here to find a file you've already uploaded](#)

- c. Enter any comments about the assignment. Then click on the *Submit Assignment* button.

File Upload [Google Doc](#)

Upload a file, or choose a file you've already uploaded.

File: Hello World.xml

square.xml

First-Expression.xml

Math-Expressions.xml

Week1LabReport

[+ Add Another File](#)

[Click here to find a file you've already uploaded](#)

3. Make sure that both lab partners have a copy of all lab files.