



Welcome – 30 course points

The purpose of this assignment is to introduce you to programming in Java and to familiarize you with the process of preparing and submitting programming assignments. You will learn how to use the *VScode* editor to edit Java files, use the command line to compile and execute your programs, as well as submit your assignment on *Autolab*.

Refer to our Programming [Assignments FAQ](#) for instructions on how to install VScode, how to use the command line and how to submit your assignments.

Programming

Write 3 programs and submit on [Autolab](#).

- [WATCH this video](#): how to open the files on VSCode and how to submit into Autolab.
- We provide a zip file (find it under Welcome on Autolab) containing *Welcome.java*, *GravitationalForce.java*, and *Palindrome.java*.
- For each problem UPDATE and SUBMIT the corresponding file.

Observe the following rules:

DO NOT add any import statements

DO NOT add the project statement

DO NOT change the class name

DO NOT change the main method

DO NOT add any new class fields

DO NOT use `System.exit()`

Observe the examples' output, display only what the problem is asking for

1. *Welcome [5 points]*. Write a program in *Welcome.java* that displays the phrase: Welcome to CS111.

- To compile the program type:
 - **javac Welcome.java**
- To test, execute the program by typing:
 - **java Welcome**

- The output will be:
 - **Welcome to CS111.**

Note: your program is expected to display **Welcome to CS111.** including the period at the end.

2. *Gravitational Force [10 points]*. Write a program in *GravitationalForce.java* that reads three input values of type double (via the command-line arguments) and displays the gravitational force (as a double value) according to the formula below. The inputs are:
- the first input is the mass of object 1 (*referred in the formula as m_1*)
 - use **Double.parseDouble(args[0])** to input the value into the program.
 - the second input is the mass of object 2 (*referred in the formula as m_2*)
 - use **Double.parseDouble(args[1])** to input the value into the program.
 - the third input is the distance between objects 1 and 2 (*referred in the formula as r*)
 - use **Double.parseDouble(args[2])** to input the value into the program.

$$F = g * (m_1 * m_2) / (r * r) \text{ where } g \text{ is } 6.6743e-11$$

Assume that the input values testing your program are greater than 0.0 (zero).

Note 1: It is not necessary to round the output value, display the value that is computed.

Note 2: Do not use ifs or loops for this program.

Note 3: DO NOT display anything other than the result value.

- To compile the program type:
 - **javac GravitationalForce.java**
- To test, execute the program by typing (use other input values to further test your program):
 - **java GravitationalForce 135.0 260.0 10.5**
- The output will be:
 - **2.1248791836734694E-8**

3. *Palindrome [15 points]*. Write a program *Palindrome.java* that takes six integer (via the command-line arguments) and prints **true** if:
- the first number is equal to the last number, and
 - the second number is equal to the second to last number, and
 - the third number is equal to the third to last number.

Prints **false** otherwise.

Use **Integer.parseInt()** to input the arguments into the program:

- use Integer.parseInt(args[0]) to input the first integer.
- use Integer.parseInt(args[1]) to input the second integer.
- use Integer.parseInt(args[2]) to input the third integer.
- use Integer.parseInt(args[3]) to input the fourth integer.
- use Integer.parseInt(args[4]) to input the fifth integer.
- use Integer.parseInt(args[5]) to input the sixth integer.

Note 1: Do not use *if* statements in this program. **Practice** writing a boolean expression.

Note 2: Assume that the inputs testing your program will always be integer numbers.

Note 3: Assume that the number of inputs testing your program will always be 6.

- To compile the program type:
 - **javac Palindrome.java**
- To test, execute the program by typing (use other input values to further test your program):
 - **java Palindrome 3 2 1 1 2 3**
- The output will be:
 - **true**

Before submission

Collaboration policy. Read our collaboration policy [here](#).

Submitting the assignment. Submit *Welcome.java*, *GravitationalForce.java*, and *Palindrome.java* separately via the web submission system called Autolab. To do this, click the *Assignments* link from the course website; click the *Submit* link for that assignment.

Getting help

If anything is unclear, don't hesitate to drop by office hours or post a question on Piazza.

- Find instructors office hours [here](#)
- Find tutors office hours in Canvas -> Tutoring ([RU CATS](#))
- Find head TAs office hours [here](#)
- In addition to office hours we have the [Coding and Social Lounge \(CSL\)](#) , a community space staffed with iLab assistants which are undergraduate students further along the CS major to answer questions.