University of Canberra

Faculty of Science and Technology

## Software Technology 2 (7170 & 9073)

Week 2 Tutorial

Algorithms – Getting Started

Tasks:

* Use the Tutorials Point website to run a Java program
* Modify an existing program on the Tutorials Point website
* Design an algorithm for a given problem
* Write a Java program to implement this algorithm
* Run and test this program on the Tutorials Point website.

Use the Tutorials Point website to run a Java program

* Open the following website <http://www.tutorialspoint.com/>
* Click Coding Ground then scroll down to Advanced IDEs at the bottom and click on Java.

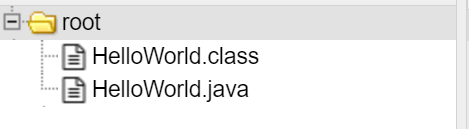
* You will see the following page (you can use this link to jump to the page: <https://www.tutorialspoint.com/online_java_compiler.php>)



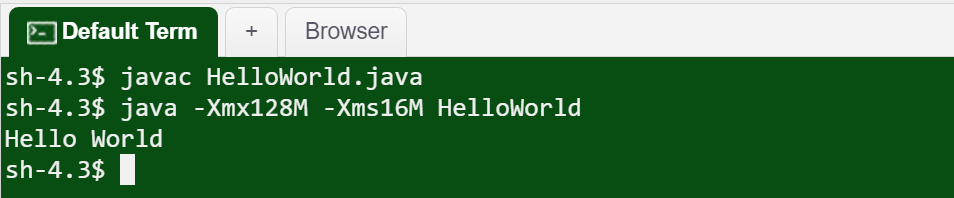
* This page contains a simple Java Console Application in the HelloWorld.java file. The program contains a line that outputs "Hello World".
* To run this program, first you click on Compile to compile the program and to check if there is error. If there is no error, you will see the following in the green area (Default Term)



javac is Java compiler which translates the program to platform-neutral Java bytecode and save it in a Java class file. Click on  to refresh the program you will see this Java class file as follows

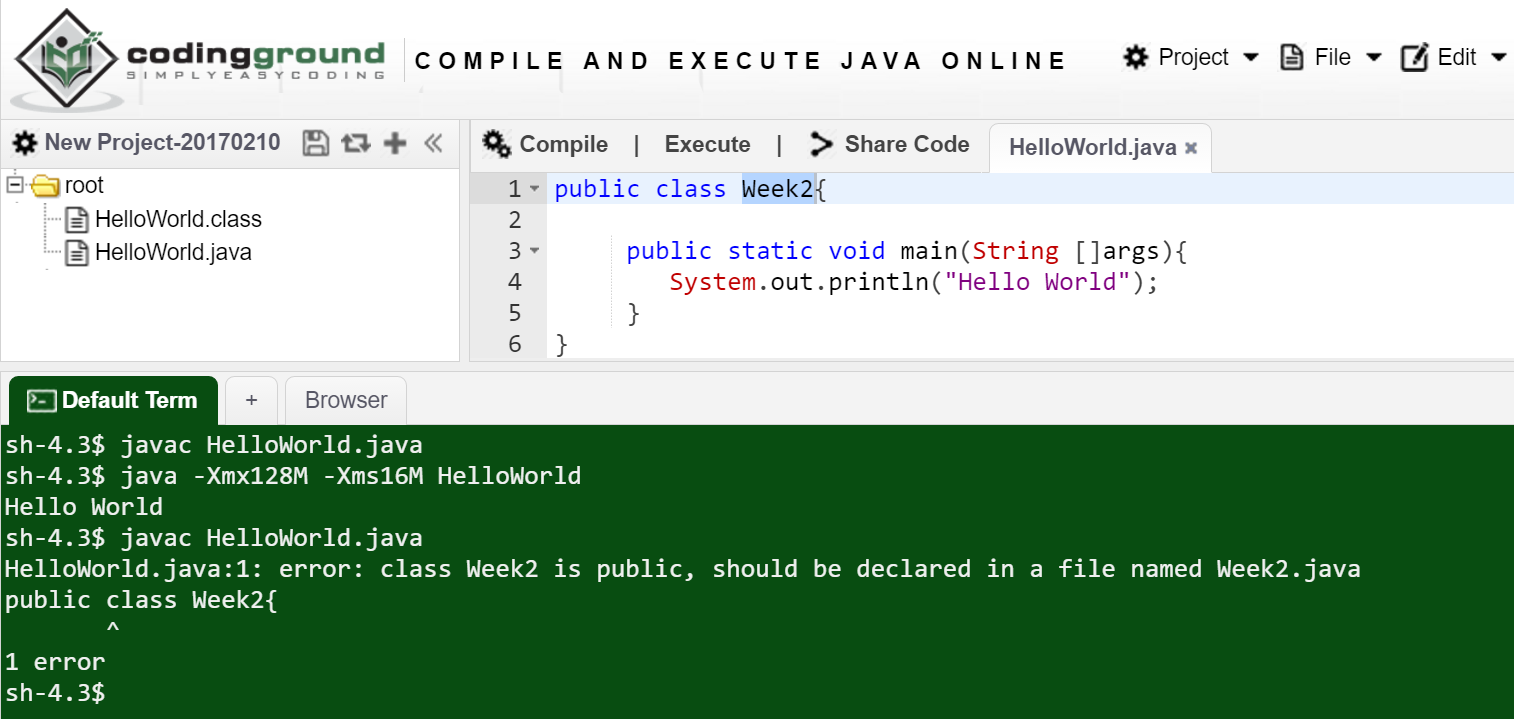


Now you click on Execute, the program runs, and outputs *Hello World* as seen below

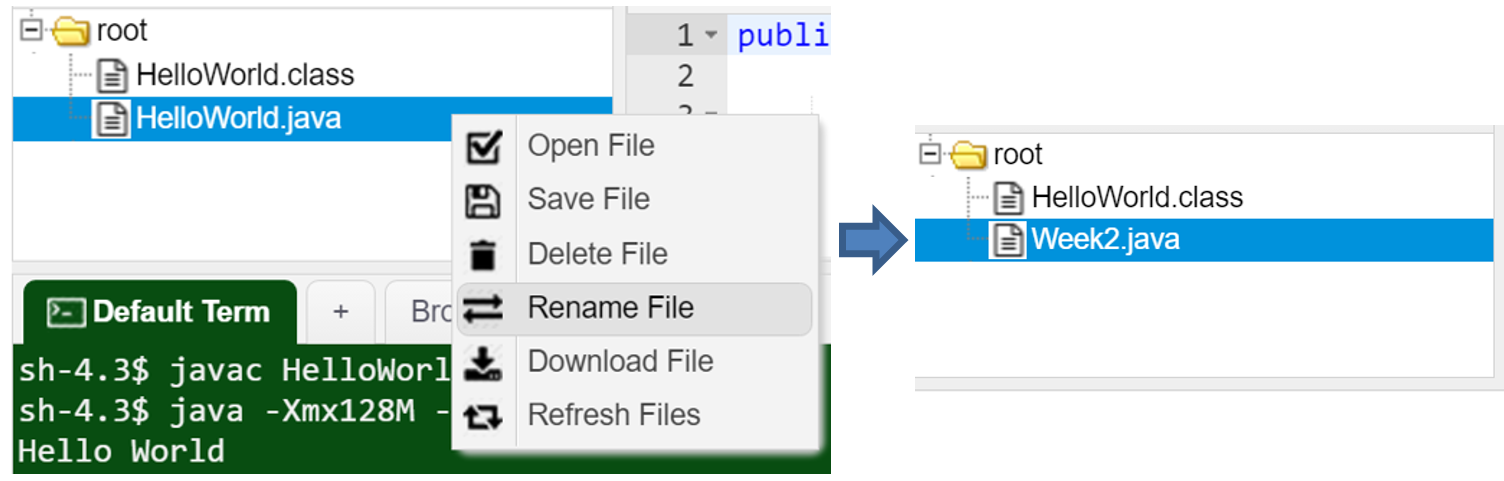


Modify an existing program on the Tutorials Point website

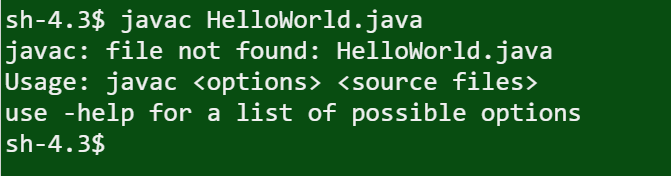
* Rename class name: in the Java program, change the class name from HelloWorld to Week2. Compile the program, you will see the error below



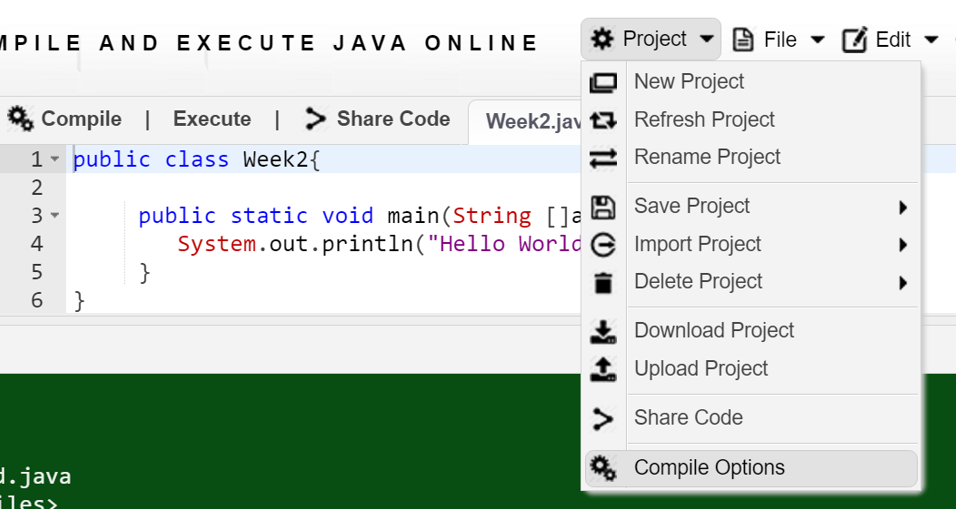
According to this error, you will want to change the filename from HelloWorld.java to Week2.java. Right click on HelloWorld.java and select Rename File and enter Week2.java.



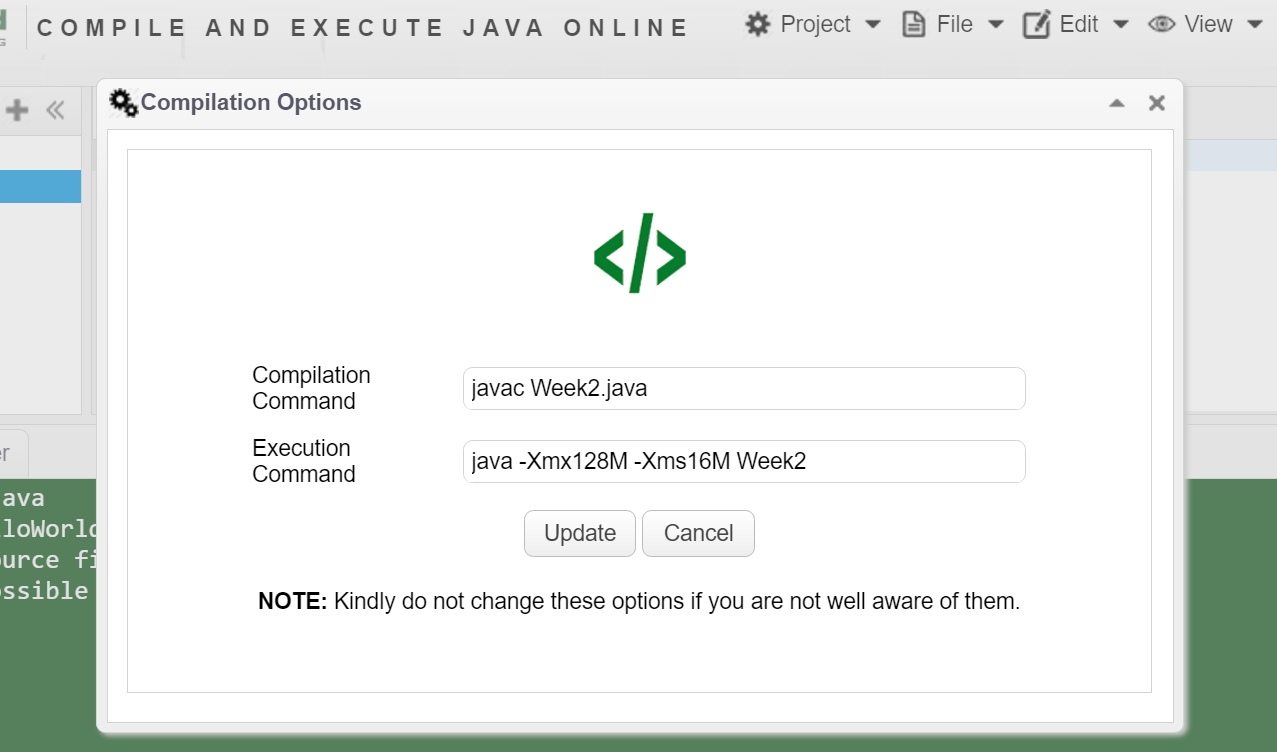
Compile the program again, you will see the following error



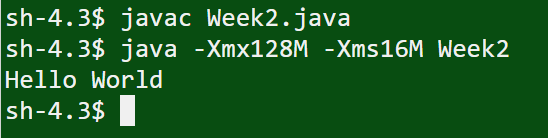
This error shows the command javac HelloWorld.java. You will need to find out this command to rename HelloWorld to Week2. Click on Project (top-right corner) and select Compile Options.



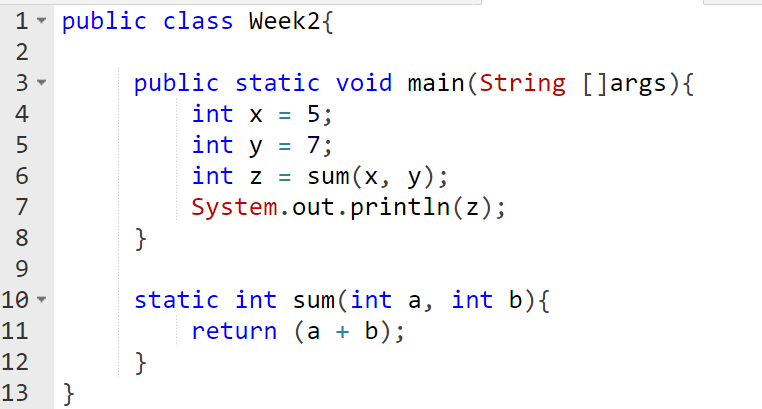
Change HelloWorld to Week2 in both Compilation and Execution commands.



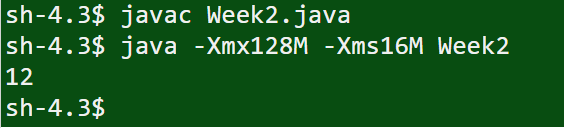
Now you compile then execute the program. The output is as follows



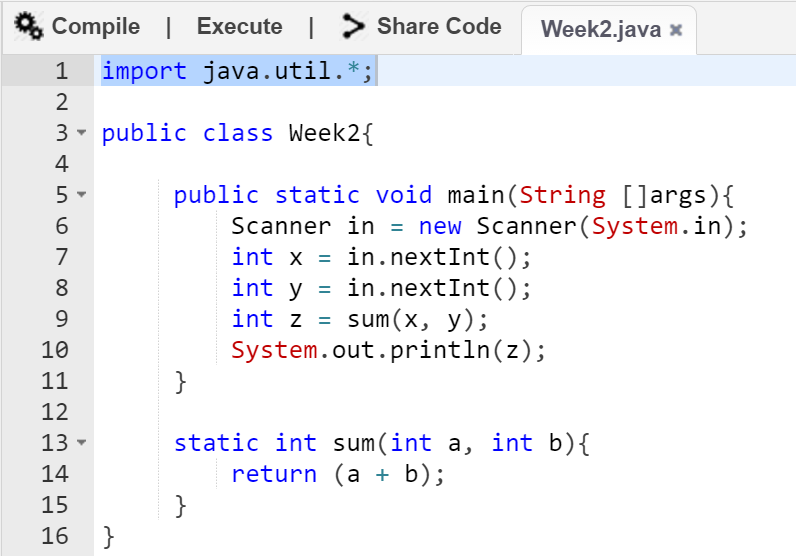
* Add a method to the program and call the method in the main method: In the screenshot below you see the sum method added to the program. This method must be in the Week2 class but not in the main method. The method should be static so you can call it in the main method as follows int z = sum(x, y); to calculate the sum of x and y then assign the sum to variable z. Modify your program according to this screenshot.



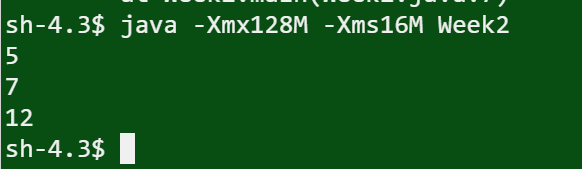
* Run (compile then execute) the program, you will see the following output



* Modify the main method so that the user can input values when the program is running: You will use the method nextInt in the Scanner class to let the user input 2 integer values and assign the first value to x and the second value to y. The program assigns the sum of x and y to variable z then outputs z. Since the Scanner class is in the java.util library so you need to import this library to the program. Use the screenshot below and change your program accordingly.



* Run (compile then execute) the program, *click on the green area* to make focus on it then enter 5 and 7 and you will see the following output



* You will use this website to run most of Java programs you implement during this semester.

Design an algorithm for a given problem

* You design an algorithm to solve the given problem and its requirements below. This problem is from Hacker Rank website.

There are N integers in an array A. All but one integer occur in pairs. Your task is to find the number that occurs only once.

Input Format: The first line contains a single integer, N, denoting the number of integers in the array. The second line contains N space-separated integers describing the respective values in array A.

Constraints

* It is guaranteed that N is an odd number.
* 1 < N < 100 (the array size is between 1 and 100, i.e. at least 1 element and at most 100 elements in the array)
* Elements in array A are integers between 0 and 100.

Output Format: Print the unique number that occurs only once in array A on a new line.

Sample 1:

Input

1

5

Output

5

*Explanation: The array only contains a single value 5, so we print 5 as our answer*

Sample 2:

Input

3

5 5 6

Output

6

*Explanation: We have two 5's and one 6. We print 6, because that's the only unique element in the array.*

* You design an algorithm for the lonelyInteger method and use it to write Java code. For example, consider the algorithm below.

1. Check if the input array has valid elements:
   * Get the number of elements in the array
   * if there is no element or more than 100 elements or the number of elements is an even number returns -1 (-1 implies wrong input)
   * If there is an element that is negative or greater than 100 returns -1
2. Finding the lonely integer in the array:
   * Start from the first element to the last in the array
   * Set a boolean variable named lonely to true
   * Find duplicates: Start from the first element to the last in the array to find if there is another element that equals to the current element.
   * If there is, set lonely to false and stop scanning (break). Otherwise set lonely to true then output the number which is the current array element

* The pseudo code for this algorithm could be as follows

int len = array length

if (len < 1 or len > 100 or len % 2 equals 0)

return -1;

endif

for i from 0 to len

if (a[i] < 0 or a[i] > 100)

return -1;

endif

endfor

for i from 0 to len

bool lonely = true;

for j from 0 to len

if (i != j) and (a[i] == a[j])

lonely = false;

break;

endif

endfor

if lonely is true

return a[i];

endif

endfor

Write a Java program to implement this algorithm

* From the pseudo code, below is the Java code for the lonelyInteger method

static int lonelyInteger(int[] a){

int len = a.length;

if (len < 1 || len > 100 || len % 2 == 0){

return -1;

}

for(int i = 0; i < len; i++){

if (a[i] < 0 || a[i] > 100){

return -1;

}

}

for (int i = 0; i < len; i++){

boolean lonely = true;

for (int j = 0; j < len; j++) {

if ((i != j) && (a[i] == a[j])) {

lonely = false;

break;

}

}

if (lonely){

return a[i];

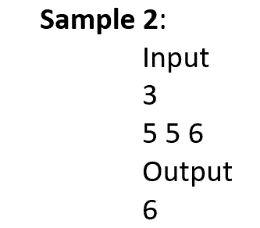
}

}

return -1;

}

* The main method should be as follows to implement the above input format

 public static void main(String[] args) {

Scanner in = new Scanner(System.in);

int n = in.nextInt();

int[] array = new int[n];

for (int i = 0; i < n; i++) {

array[i] = in.nextInt();

}

int x = lonelyInteger(array);

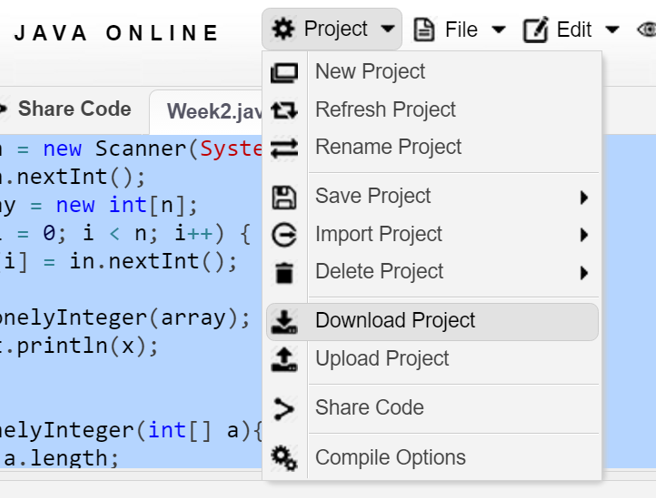
System.out.println(x);

}

Run and test this program on the Tutorials Point website

* Delete the current main method and the sum method in the Week2 class. Add the main method above and the lonelyInteger to the Week2 class.
* Compile and execute the program.
* Enter sample input 1 above and see what your program outputs and then compare it with the sample output 1 above.
* Enter sample input 2 above and see what your program outputs and then compare it with the sample output 1 above.

Download your program and submit it via Canvas site of this unit and let your tutor see your complete work before you leave the lab. Mark for class attendance: 1 and mark for this program: 1. See the screenshot below to know how to download your program.



Additional algorithm for the lonelyInteger method (not for assessment)

* Below is another solution to the lonelyinteger method. Test it and try to understand it

static int lonelyInteger(int[] a) {

int len = a.length;

if (len < 1 || len > 100 || len % 2 == 0){

return -1;

}

for(int i = 0; i < len; i++){

if (a[i] < 0 || a[i] > 100){

return -1;

}

}

// Start finding the lonely integer using XOR

// XOR "exclusive or" is a bitwise (and logical) operation

// that outputs 1 (true) if the two inputs are not the same

// and outputs 0 (false) if the two inputs are the same

// x XOR 0 = x

// x XOR y XOR y = x XOR (y XOR y) = x XOR 0 = x

int x = 0;

for (int i = 0; i < len; i++) {

x ^= a[i];

}

return x;

}