

Tutorial For Basic Use of IntelliJ IDEA and Input Output

Based on the tutorial of "2020S-Java-A" designed by teaching group in SUSTech

Modified (mainly change to markdown file) by ZHU Yueming in 2021. Jan. 11th

Modified by WANG Daxing in 2022. Sept.

Objectives

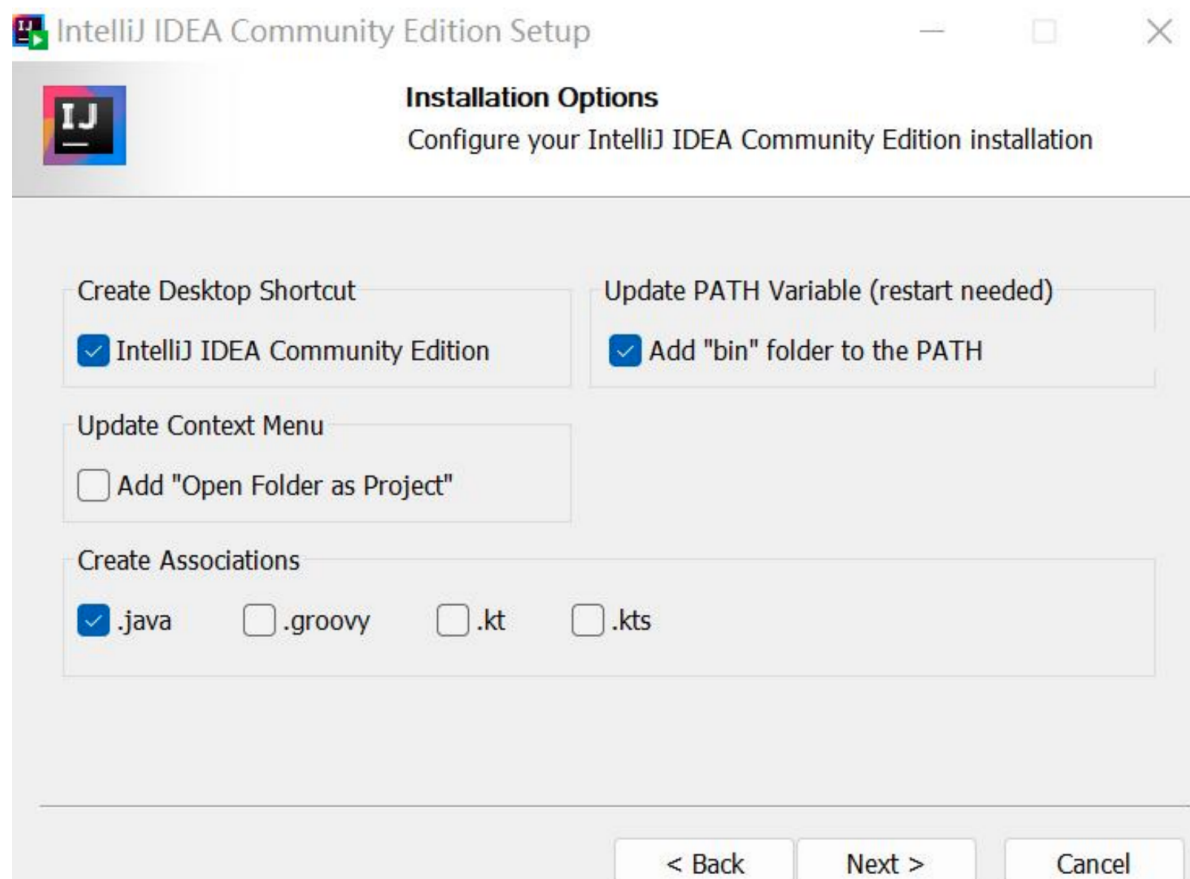
1. Learn how to use an **Integrated Development Environment (IDE)** in writing JAVA programs
2. Practice using input and output statements.
3. Practice storing values with primitive types

Software Installation

In this course, we will use IDEA as our reference IDE.

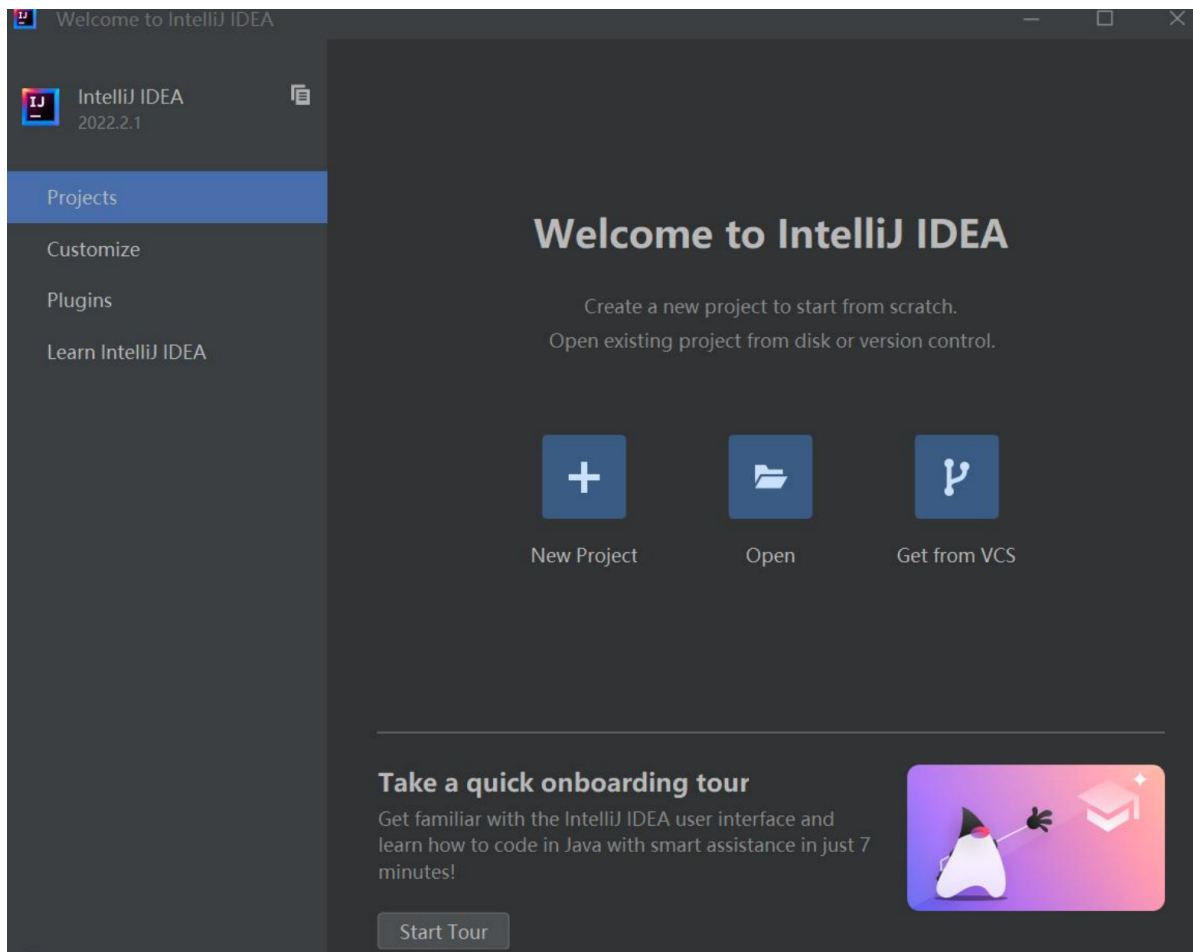
You can download IDEA (community version) at the following link: <https://www.jetbrains.com/idea/download/>

Once downloaded, run the executable. Follow the prompts to install IDEA, tick "Add launchers dir to the PATH" and ".java in Create Associations" as follows:

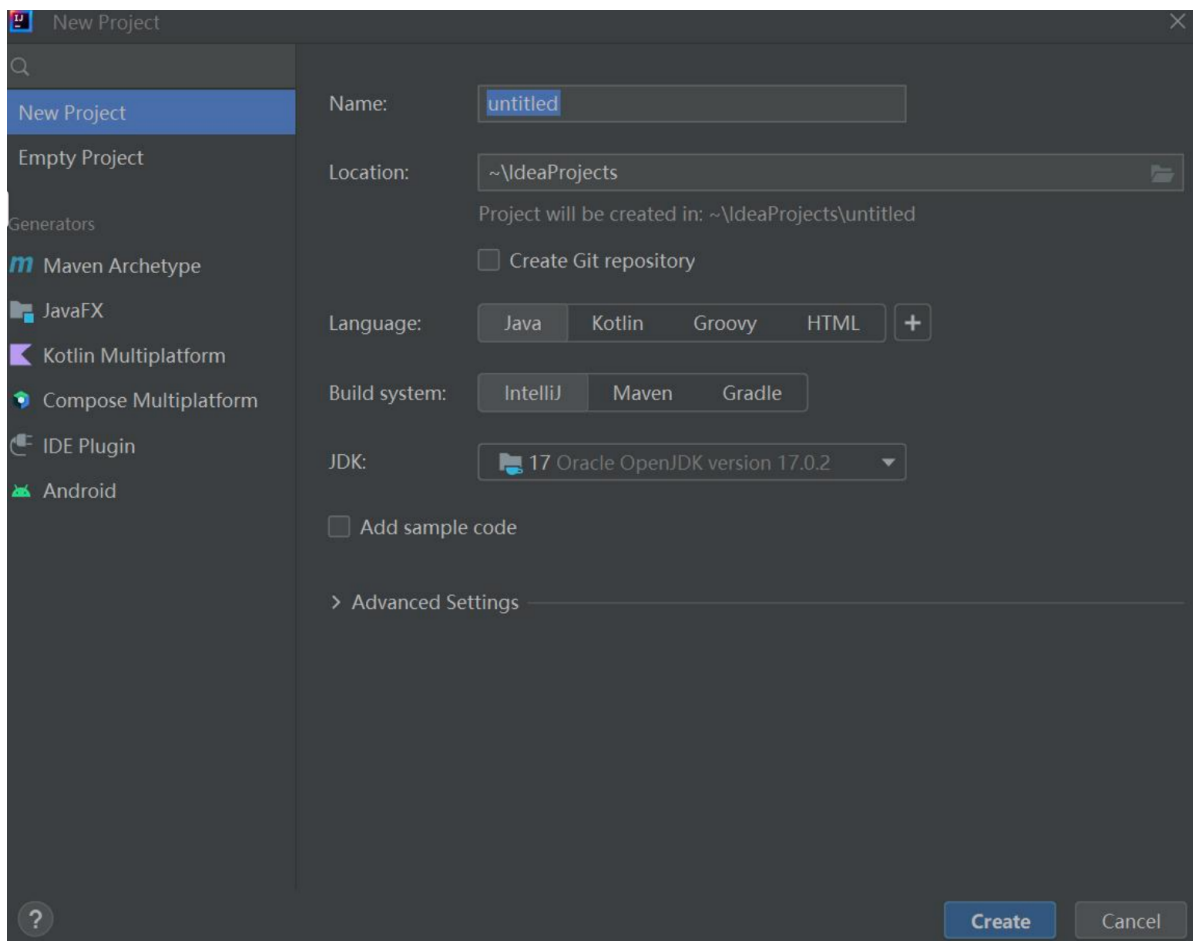


After the installation, you are suggested to restart your computer.

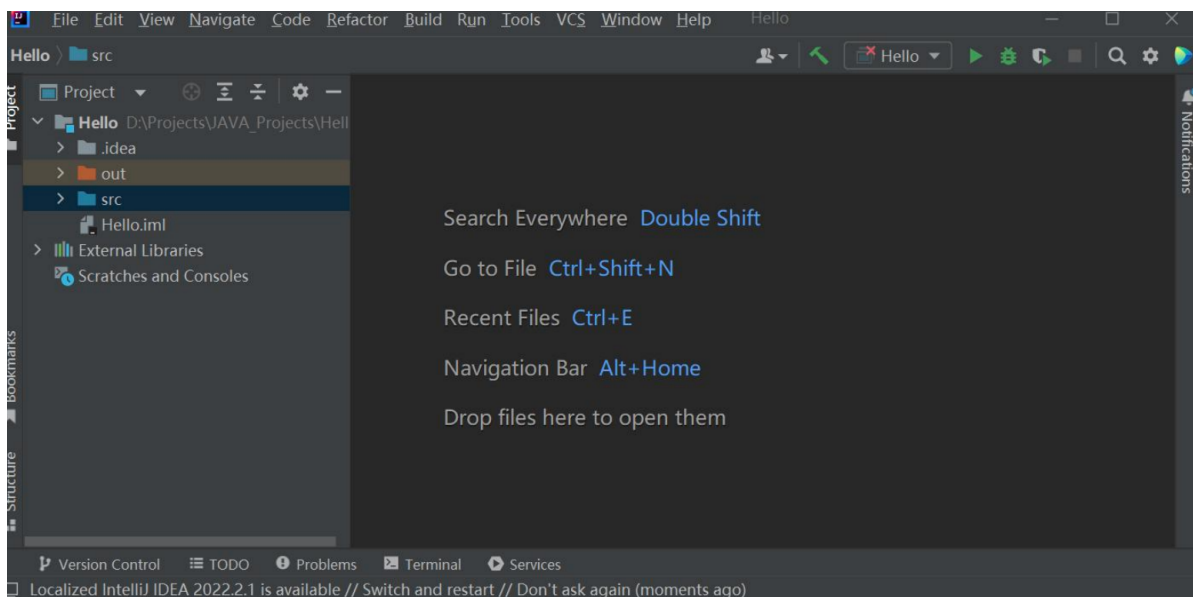
Find your IntelliJ IDEA Community Edition in the Start menu to start IDEA. Confirm the Privacy Policy (You are recommended to read the policy for this and any other software), and decide whether you want to send anonymous statistics data to the software developer, i.e., JetBrains. Finally, the following start window prompts:



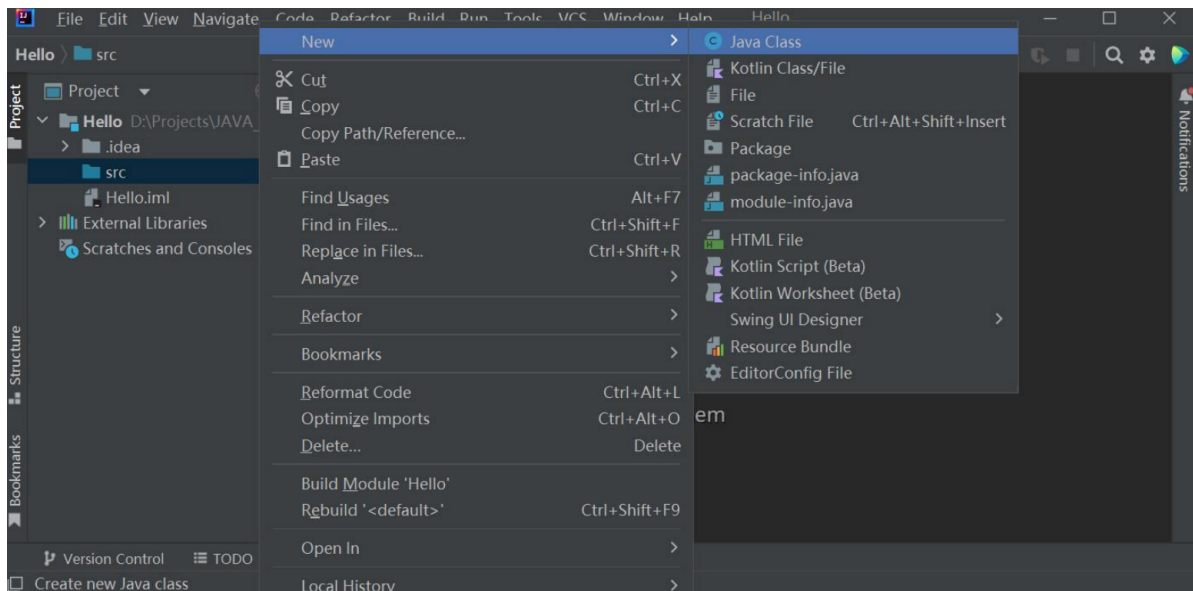
To create a new java program, click New Project. Set project name and location. This is where you want to put all your project files. Make sure that the Project SDK at the top is set. Then press "Create" with the default setting.



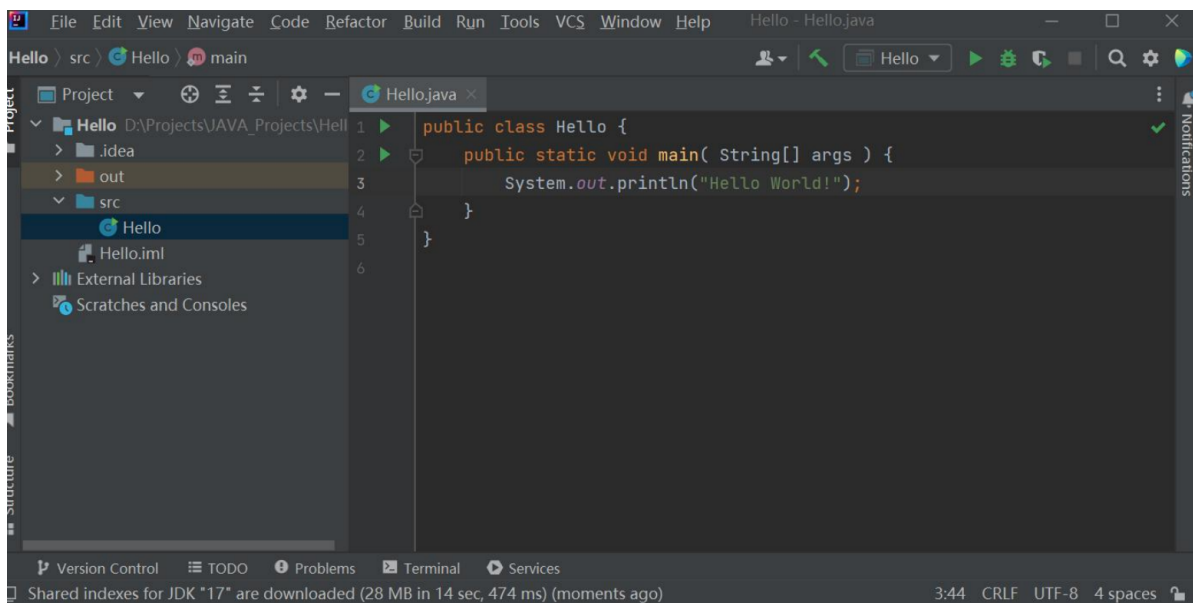
The project should appear and look like follows:



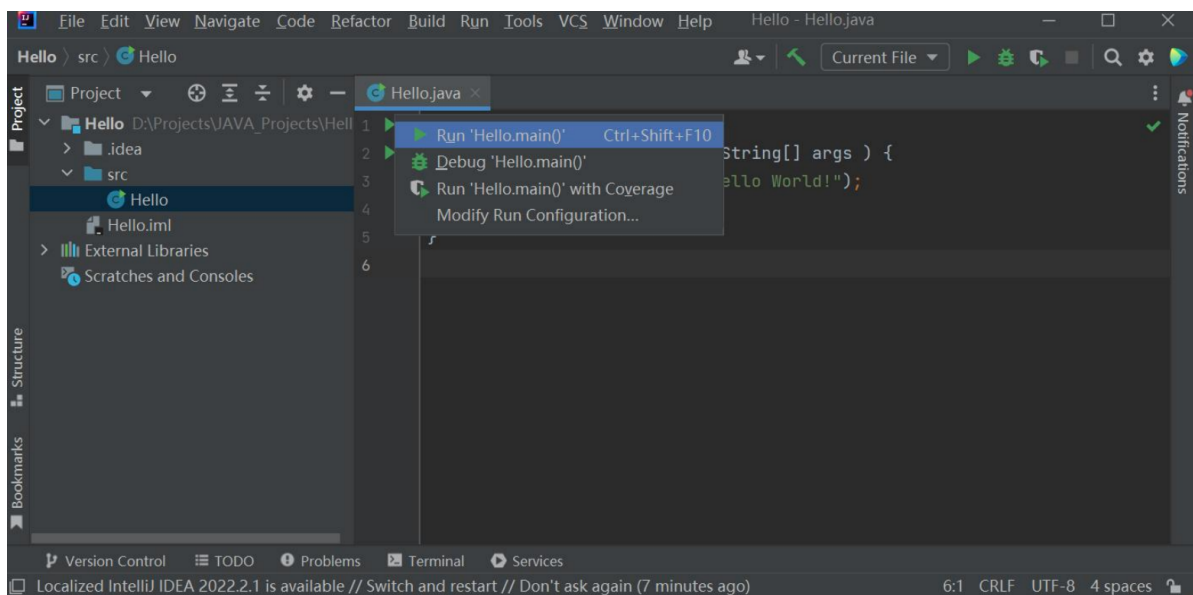
In order to start here, a java file should be created. Right click "src" (which corresponds to the "src" directory in the project directory just set), choose "new" and "Java Class", left click "Java Class" to create a java file.



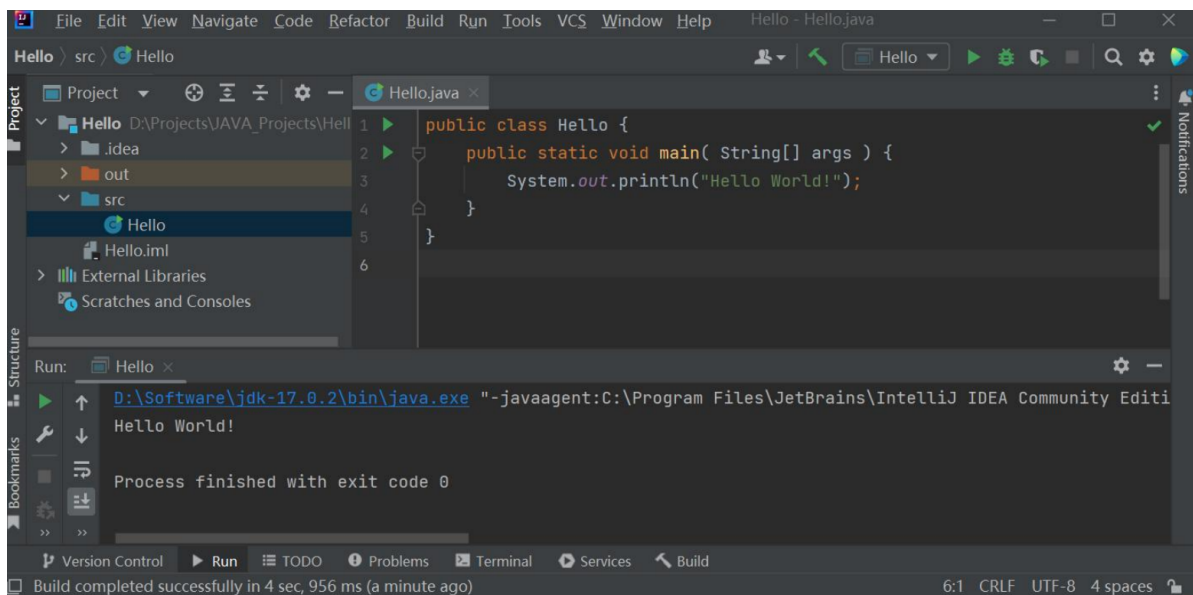
After the java file is created, write some code here:



Then build and run the code, simply click the green triangle:



The following result should be seen:



2 Exercise

2.1 Exercise 1

Write and test the following source code to see how Scanner class works:

```
import java.util.Scanner;  
  
public class Sum {  
    public static void main(String[] args) {  
        System.out.println("welcome to CS109!");  
  
        Scanner input = new Scanner(System.in);  
  
        int number1, number2, sum;  
  
        System.out.print("Enter the first integer: ");  
        number1 = input.nextInt();  
        System.out.print("Enter the second integer: ");  
        number2 = input.nextInt();  
  
        sum = number1 + number2;  
        System.out.printf("Sum is %d\n", sum);  
    }  
}
```

2.2 Exercise 2

Write a program that prompts the user to enter his information, and then prints out in a specific format:

```
import java.util.Scanner;  
  
public class Information {  
    public static void main(String[] args) {  
        String name;  
        int age;  
        float weight;  
        char grade;
```

```
// Creating object of Scanner class
Scanner input = new Scanner(System.in);

System.out.print("Enter your name: ");
name = input.next();
System.out.print("Enter your age: ");

age = input.nextInt();
System.out.print("Enter your weight in KG: ");
weight = input.nextFloat();
System.out.print("Enter your highest grade in last semester: ");
grade = input.next().charAt(0);

System.out.printf("You are %s.\nYou are %d years old.\n", name, age);
System.out.printf("You weigh %.1f KG.\nThe highest grade you got is
%c\n", weight, grade);
    }
}
```

The output looks like this:

```
Enter your name: Jack
Enter your age: 20
Enter your weight in KG: 60.5
Enter your highest grade in last semester: A
You are Jack.
You are 20 years old.
You weigh 60.5 KG.
The highest grade you got is A
```

What happens if you enter 21.5 to the age? Try it out. We will talk about exception handling later in this course.

2.3 Exercise 3

Write a program that prompts the user to enter the height and width of a rectangle then prints the area and perimeter of the rectangle. The area and perimeter should be printed to the nearest two decimal place. The output looks like this:

```
Enter the width of a rectangle: 1.7
Enter the height of a rectangle: 2.4
The area is 4.08
The perimeter is 8.20
```

2.4 Exercise 4

Write a time converter that prompts the user to enter the number of seconds then prints the equivalent time in hours, minutes and seconds. The output looks like this:

```
Enter the number of seconds: 7402
The equivalent time is 2 hours 3 minutes and 22 seconds.
```

