UNIVERSITY OF MISSOURI-COLUMBIA
COLLEGE OF ENGINEERING
DEPARTMENT OF ELECTRICAL ENGINEERING AND COMPUTER SCIENCE

**3330 – OBJECT-ORIENTED PROGRAMMING**

# Laboratory 1: Getting started & IntelliJ IDE

**(No late submissions will be accepted for this Lab)**

## 1 Goal

In this lab you will learn how to use IntelliJ IDEA to create, edit, and run simple Java programs. Java and IntelliJ IDEA IDE commonly used by developers both in academia and in industry, so your familiarity with them will be a valuable skill. It is recommended to use your own personal computer (laptop) to do this and the feature labs.

## 2 Resources

- Lecture notes "Unit 1"

## 3 Directed Lab Work

### 3.1 Install and Register the IntelliJ IDE

- We recommend using free individual student/teacher licenses. Students creating an account with @missouri.edu email address can instantly get a free subscription. https://www.jetbrains.com/shop/eform/students
- Download IntelliJ IDEA "Ultimate" from the link: https://www.jetbrains.com/idea/download/.

### 3.2 Your First Java Program

Java is primarily an object-oriented language. It means a Java program can be a collection of objects that communicate via calling each other's methods. A typical Java program includes a lot of classes, interfaces, objects, and other concepts from object-oriented programming.

Java program should have at least one class and the main method inside it to start the program. The main method is the entry point for any applications.

#### 3.2.1 The declaration of the main method

The main method is the entry point of any Java programs. It has a very specific syntax which you need to remember. Let's see an example of the simplest application that prints the text "Hello, Java" in the standard output:

```
class Task {
  public static void main(String[] args) {
    System.out.println("Hello, Java");
  }
}
```

Here is a class named *Task*. The class contains the main method for starting the program. It is important to mention that a class containing the main method can have any name, but the main method should always have the same name "main".

Let's take a closer look at the declaration of the main method:

```
  public static void main(String[] args) {
```

- The keyword ***public*** indicates that the method can be invoked from everywhere
- The keyword ***static*** indicates the method can be invoked without creating an instance of the class
- The keyword ***void*** indicates the method doesn't return any value
- The array variable ***args*** contains arguments entered at the command line if there are no arguments then the array is empty.

As you can see, even the simplest Java application contains a lot of concepts. All of them will be studied during the future lectures and lab exercises. For now, you just need to understand how to write and run a simple Java program with the main method.

### 3.2.2 Task 1

- Use IntelliJ IDE to write a simple Java program that includes the main method. Your **project** must be named **Task1**. See Figure 1. The program should print the following text: "Welcome to 3330 course!".
- Select the latest JDK which will be "openjdk-24 Oracle Open JDK 24.0.2+12". Download this version if not yet ready in your IDE.
- Run your program and make sure it prints the message "Welcome to 3330 course!" in the IntelliJ IDE run window.

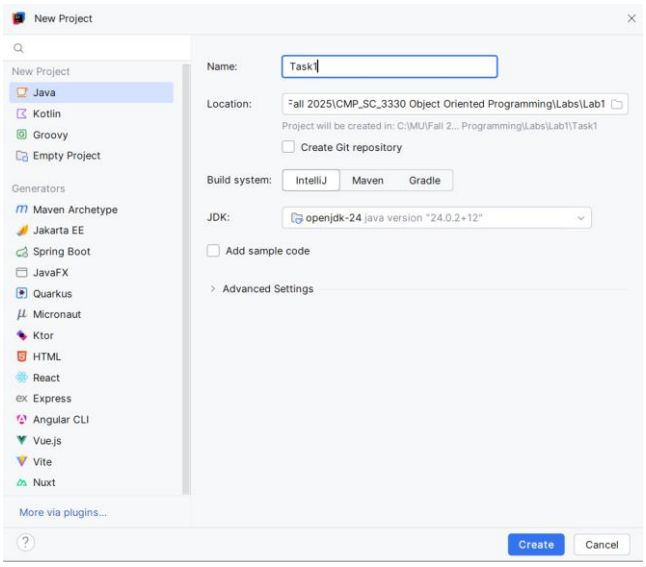


Figure 1

## 3.3 Use multiple println methods

The ***println*** method can be repeated number of time to display simple patterns. For example, if we need to display on the screen the following triangle pattern,

```
  *
 * *
* * *
```

we can use three consequence ***println*** methods to do so as shown in the following code. Notice the white spaces added before or in between the starts to make the pattern.

```
class Task2 {
  public static void main(String[] args) {
    System.out.println("  *");
    System.out.println(" * *");
    System.out.println("* * *");
  }
}
```

### 3.3.1 Task 2

- Using IntelliJ IDE create a new project (using the menu option File → new Project) and named it **Task2**.
- Create a new Java class and name it Task2, write a main method in this class.

```
   *     *     *     *     *
   *    * *    *    *    * *
*  *  *****    * *    *****
 **  *     *    *    *     *
```

Figure 2

- In the main method write four consequence *println* methods to display the pattern shown in Figure 2.
- Run your program and make sure it prints the exact pattern shown in Figure 2.
- Don't forget to add comments.

## 4 Hand In

- Create a new folder and name it **yourUMId_Lab1**, then copy the folders of projects **Task1** and **Task2** in your new folder **yourUMId_Lab1**. yourUMId is the first part of your UM (University of Missouri) email address before the @ sign. Zip yourUMId_Lab1 folder into yourUMId_Lab1.zip.

- **Please note that the naming conventions, the types, and the submitted file structure are very important. The improper file structures, types or names will cause losing marks.**
- 
- Submit your *yourUMId_* Lab1.zip file into Canvas on or before **September 5, 4:59 PM**.
- **No submissions will be accepted after September 5, 5:00 PM**