

## **Lab 1 – Java Introduction (I)**

---

**The Objective of this Lab is to understand the basics of Java Syntax which includes:**

**1.1 The Eclipse IDE(Optional)**

**1.2 Java Input/output**

**1.3 Java Conditional Statements**

**1.4 Java Arrays & Loop Controls**

## **1.1 The Eclipse IDE**

### **Using Terminal/Command Line**

You can download the latest version of javacc from <https://javacc.github.io/javacc/>

Setup the PATH environment variable.

Use the following commands to compile your grammar file Example.jj:

1. Run javacc on the grammar input file to generate a bunch of Java files that implement the parser and lexical analyzer (or token manager):

```
javacc Example.jj
```

2. Now compile the resulting Java programs:

```
javac *.java
```


3. The parser is now ready to use. To run the parser, type:

```
java Example
```

### **Using IDE**

#### **Opening the New Java Project wizard**

The New Java Project wizard can be used to create a new java project. There are many ways to open this wizard:

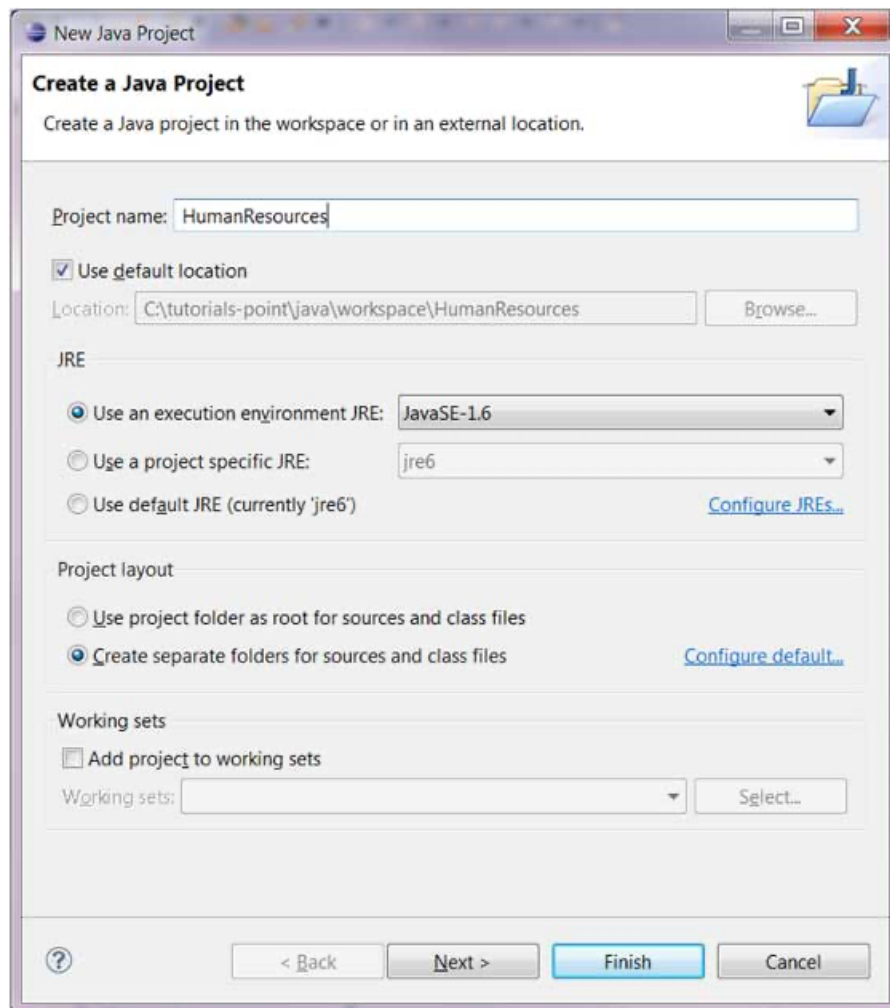
- By clicking on the File menu and choosing New > Java Project
- By right clicking anywhere in the Project Explorer and selecting New > Java Project
- By clicking on the New button (  ) in the Tool bar and selecting Java Project

#### **Using the New Java Project wizard**

The New Java Project Wizard has two pages.

On the first page:

- Enter the Project Name
- Select the Java Runtime Environment (JRE) or leave it at the default
- Select the Project Layout which determines whether there would be a separate folder for the sources code and class files. The recommended option is to create separate folders for sources and class files.

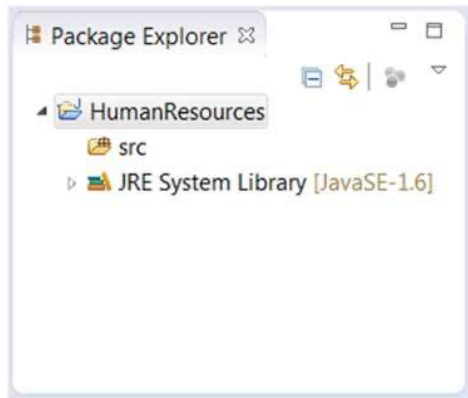


You can click on the Finish button to create the project or click on the Next button to change the java build settings.

On the second page you can change the Java Build Settings like setting the Project dependency (if there are multiple projects) and adding additional jar files to the build path.

### Viewing the newly created project

The package explorer shows the newly created Java project. The icon that represents a Project is decorated with a "J" to show that it is a Java Project. The folder icon is decorated to show that it is a java source folder.



## 1.2 Java Input/output

When we consider a Java program it can be defined as a collection of objects that communicate via invoking each other's methods. Let us now briefly look into what do class, object, methods and instance variables mean.

- **Object** - Objects have states and behaviors. Example: A dog has states - color, name, breed as well as behaviors -wagging, barking and eating. An object is an instance of a class.
- **Class** - A class can be defined as a template/ blue print that describes the behaviors/states that object of its type support.
- **Methods** - A method is basically a behavior. A class can contain many methods. It is in methods where the logics are written, data is manipulated and all the actions are executed.
- **Instance Variables** - Each object has its unique set of instance variables. An object's state is created by the values assigned to these instance variables.

### First Java Program:

Let us look at a simple code that would print the words *Hello World*.

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

Save the above file as Simple.java.

To compile:            javac Simple.java

To execute:            java Simple

Output:

Hello World

## 1.3 Java Conditional Statements

- **If/else Statement**

```
public class Test {  
  
    public static void main(String args[]) {  
        int x = 30;  
  
        if ( x < 20 ) {  
            System.out.print("This is if statement");  
        } else {  
            System.out.print("This is else statement");  
        }  
    }  
}
```

- **Switch Statements**

```
public class Test {  
    public static void main(String args[]) {  
        char grade = 'C';  
  
        switch(grade)  
        {  
            case 'A' :  
                System.out.println("Excellent!");  
                break;  
            case 'B' :  
            case 'C' :  
                System.out.println("Well done");  
                break;  
            case 'D' :  
                System.out.println("You passed");  
            case 'F' :  
                System.out.println("Better try again");  
                break;  
            default :  
                System.out.println("Invalid grade");  
        }  
        System.out.println("Your grade is " + grade);  
    }  
}
```

## 1.4 Java Loop Controls

### **For Loop:**

```

public class TestArray {
    public static void main(String[] args) {
        double[] myList = {1.9, 2.9, 3.4, 3.5};

        // Print all the array elements
        for (int i = 0; i < myList.length; i++) {
            System.out.println(myList[i] + " ");
        }
        // Summing all elements
        double total = 0;
        for (int i = 0; i < myList.length; i++) {
            total += myList[i];
        }
        System.out.println("Total is " + total);
        // Finding the largest element
        double max = myList[0];
        for (int i = 1; i < myList.length; i++) {
            if (myList[i] > max) max = myList[i];
        }
        System.out.println("Max is " + max);
    }
}

```

### While Loop:

```

class whileDemo {
    public static void main(String[] args){
        int count = 1;
        while (count < 11) {
            System.out.println("Count is: " + count);
            count++;
        }
    }
}

```

### Do-While Loop:

```

public class Test {
    public static void main(String args[]){
        int x = 10;

        do{
            System.out.print("value of x : " + x );
            x++;
            System.out.print("\n");
        }while( x < 20 );
    }
}

```

### Lab Tasks:

- 1) Make a simple Calculator in Java that takes two number from the user and asks the user to choose any operation (+ - \* and /) to be performed on the numbers.
- 2) Take an array of five elements and find the Minimum and Maximum number out of it. Do not use library functions, right your own.
- 3) Enter the Marks of a student for different subjects. Print the corresponding Grades.