CS 102
Object Oriented Programming

Hello World

Reyyan Yeniterzi
reyyan.yeniterzi@ozyegin.edu.tr

September 19, 2016

Hello World Application

- Develop a Hello World application

Hello World Application

- Develop a Hello World application
  - We need to implement the HelloWorld class.

```
public class HelloWorld {


}
```

Hello World Application

- Develop a Hello World application
  - We need to implement the HelloWorld class.

```
public class HelloWorld {


}
```

Hello World Application


```
public class HelloWorld {


}
```

- Access modifier
  - They are used to set access levels for classes, variables, and other entries.

Hello World Application


```
public class HelloWorld {


}
```

- Access modifier
  - They are used to set access levels for classes, variables, and other entries.
  - For the top level classes it can be either
    - **public** or
    - **default** (no keyword)

Hello World Application


```
public class HelloWorld {


}
```

- When it is public, this class is visible to the earth and can be accessible from everywhere.

Hello World Application


```
package week1;

class HelloWorld {

}
```

- Without any access modifier this class is visible only within package Week1 and can be accessible from package Week1 only.
- This default access modifier is also known as **package-private**.

Hello World Application


```
package week1;

class HelloWorld {

}
```

- Without any access modifier this class is visible only within package Week1 and can be accessible from package Week1 c
- This default access
  package-private.

Package is a namespace to organize a set of related classes

Hello World Application


```
private class HelloWorld {


}
```

- A top level class cannot be private.
- Illegal modifier for the class HelloWorld

- Access modifiers will be covered in detail later in the course…

Hello World Application


```
public class HelloWorld {


}
```

- The keyword class for the class definition

Hello World Application


```
public class HelloWorld {


}
```

- The name of the class

Hello World Application


```
package week1;

class HelloWorld {
    public static void main(String args[]) {

    }
}
```

- No **run** method, instead we have the **main** method
- The entrance point of our program

Hello World Application


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

}
```

- This signature is always the same.

Hello World Application


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

}
```

- Main method accepts a single argument
  - An array of element of type String
  - This array holds the **command-line arguments**

```
java HelloWorld arg1 arg2
```

Hello World Application


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

Command-line arguments are passed to the program from the command line following the name of the program. They are like parameters of the program.

- M
  - A
  - This array holds the **command-line arguments**

```
java HelloWorld arg1 arg2
```

Hello World Application


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

}
```

- The access modifier
- It is **public**, therefore can be reached from anywhere.

Hello World Application


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

}
```

- This method is static means that we can call this method without creating an object from it.
- Since this is the first method being called during the execution of the program, no object has been instantiated yet.
- More details of this will be described later in this course.

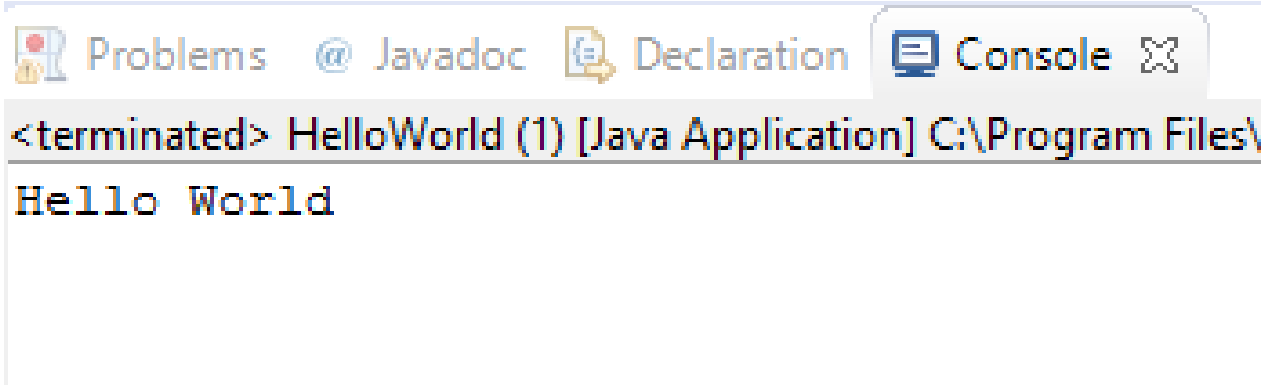
Hello World Application


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

}
```

- The return type of the method.
- Nothing to return, since there is not a single line of code waiting for this function's return.
- But, the main function may return some platform specific values which indicate the final status of the program.
  - You don't have to worry about these.

Hello World Application


```
package week1;

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



Hello World Application


```
package week1;

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

- Sending **println** message to the **out** field of the **System** class which is the standard output.

Practice

- More details regarding Java SDK is available at 12-Java_SDK.pdf from CS101 lecture notes.

- Remove the ACM library from your CS101 assignments and make the necessary changes!

Any Questions ?