University of
HUDDERSFIELD
Inspiring tomorrow's professionals

Software Design & Development
CFS2160

Week 9 – Creating a Java Class in IntelliJ

Session Plan

- An important message
- Firstly we are going to look at class structure in Java and how to create a class in IntelliJ
- Work on todays worksheet.
- Then work on any outstanding practical work.

Please Note

We Only Use Java Version 8!

If you create your second coursework in a version newer than V8 it WILL NOT work at uni!

Anatomy of a Java Class

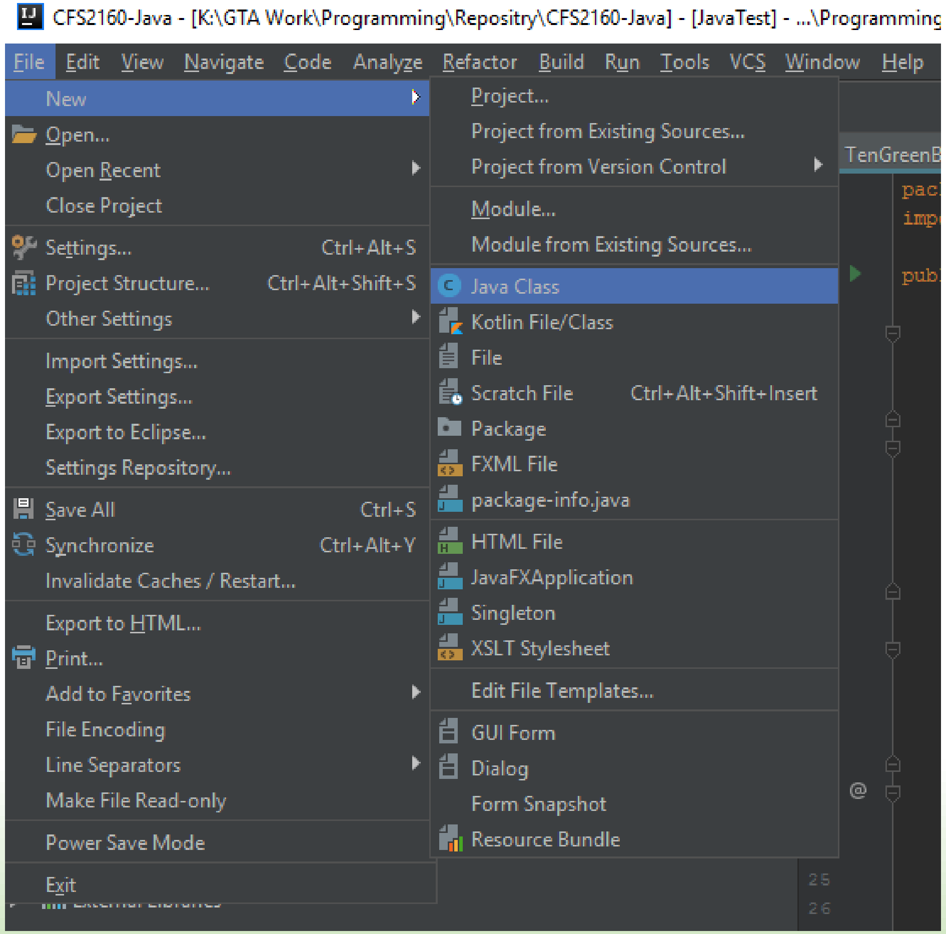
Java classes require most but not all of the following items to be used in a programme. Generating the following when creating a new class is a very good habit to get into.

- Class Name (must be the same as the file name without exception!)
- Attributes (which require a datatype, int, String etc. Should always be private)
- Constructor (method to create and initialise the state of an object of a class)
- Methods (perform other logic in the program)
- Getters and Setters (public methods used to get and set the values of attributes)
- Main Method (used as the starting point of the program, not always required)

IntelliJ contains the functionality for us to create EVERYTHING we need for a class.

Create a Class

In IntelliJ select, ‘File > New > Java Class’



Make sure your class / file name is sensible and relates to the real world object it represents.

Class names should start with a capital, each subsequent word should be capitalised.

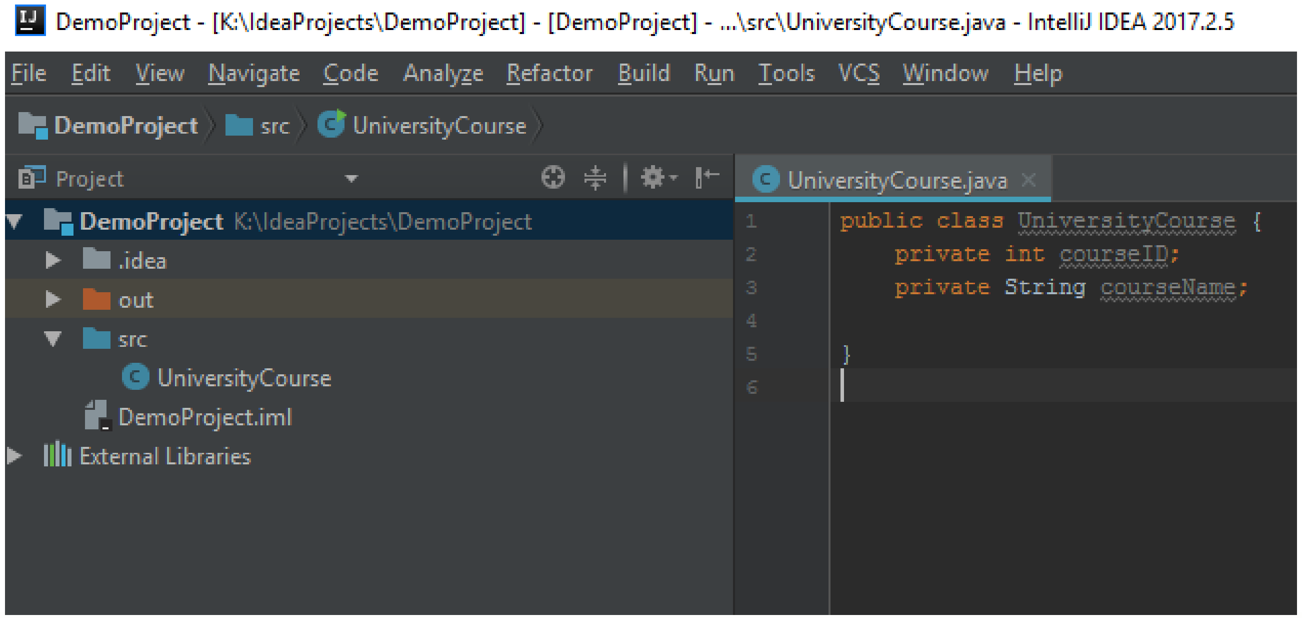
i.e.

UniversityCourse

Tip, it is a good idea to think about folder names and structure when creating a new file.

Add Attributes

Add any attributes below the class name, use sensible names, the first letter should be lowercase and each subsequent word capitalised, you can add more attributes later if required.


DemoProject - [K:\IdeaProjects\DemoProject] - [DemoProject] - ...\src\UniversityCourse.java - IntelliJ IDEA 2017.2.5
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
DemoProject src UniversityCourse
Project
UniversityCourse.java
DemoProject K:\IdeaProjects\DemoProject
.idea
out
src
UniversityCourse
DemoProject.iml
External Libraries

```java
public class UniversityCourse {
    private int courseID;
    private String courseName;

}
```

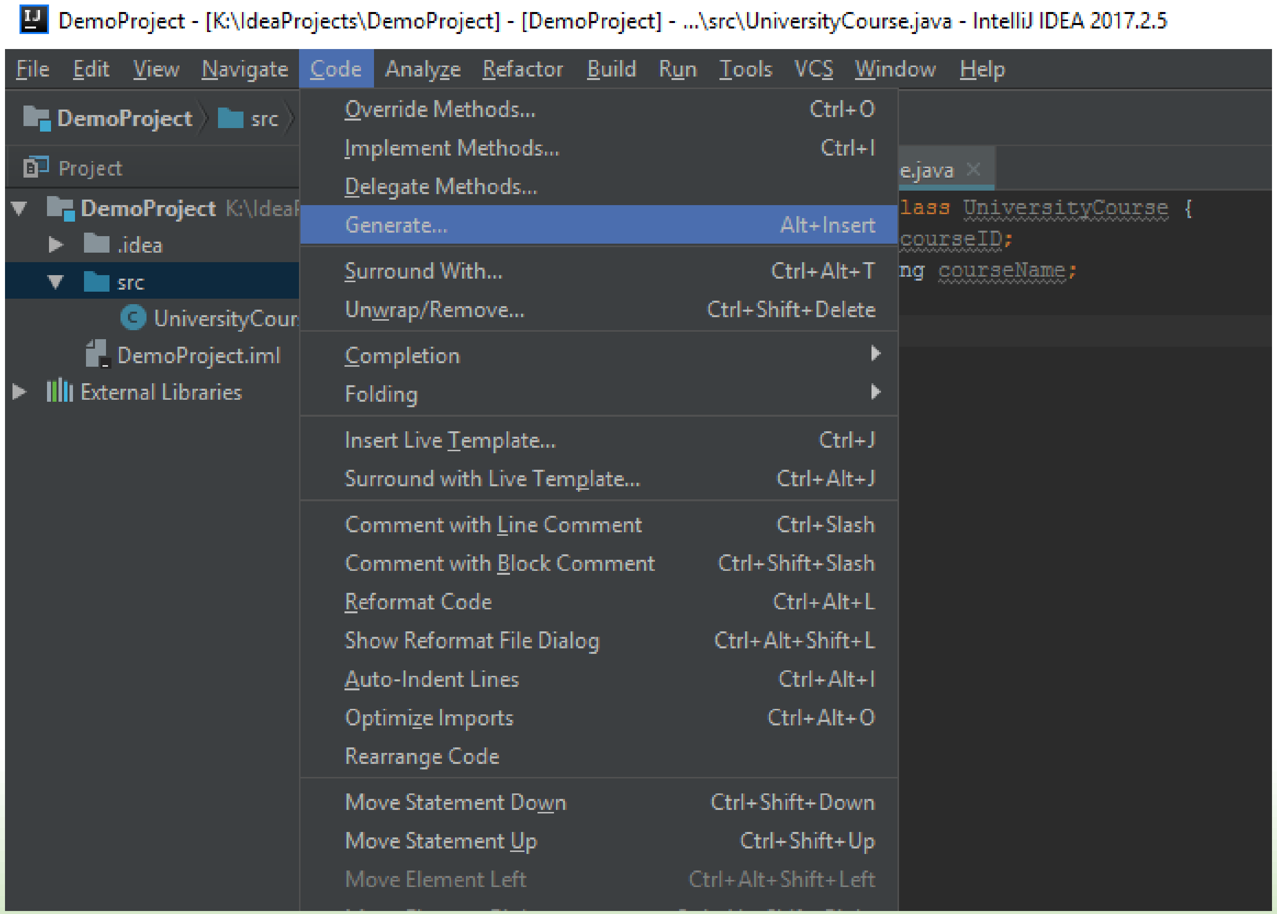
In Java, we have to declare the data type of the attribute.

We also have to set its access modifier, should be private.

We also need to the ; symbol at the end of each definition.

Add Getters and Setters

To generate the Get and Set methods in IntelliJ click ‘Code > Generate’


DemoProject - [K:\IdeaProjects\DemoProject] - [DemoProject] - ...\src\UniversityCourse.java - IntelliJ IDEA 2017.2.5
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
DemoProject src
Project
DemoProject K:\IdeaP
.idea
src
UniversityCour
DemoProject.iml
External Libraries
e.java
Override Methods... Ctrl+O
Implement Methods... Ctrl+I
Delegate Methods...
Generate... Alt+Insert
Surround With... Ctrl+Alt+T
Unwrap/Remove... Ctrl+Shift+Delete
Completion
Folding
Insert Live Template... Ctrl+J
Surround with Live Template... Ctrl+Alt+J
Comment with Line Comment Ctrl+Slash
Comment with Block Comment Ctrl+Shift+Slash
Reformat Code Ctrl+Alt+L
Show Reformat File Dialog Ctrl+Alt+Shift+L
Auto-Indent Lines Ctrl+Alt+I
Optimize Imports Ctrl+Alt+O
Rearrange Code
Move Statement Down Ctrl+Shift+Down
Move Statement Up Ctrl+Shift+Up
Move Element Left Ctrl+Alt+Shift+Left

```
lass UniversityCourse {
courseID;
ng courseName;
```

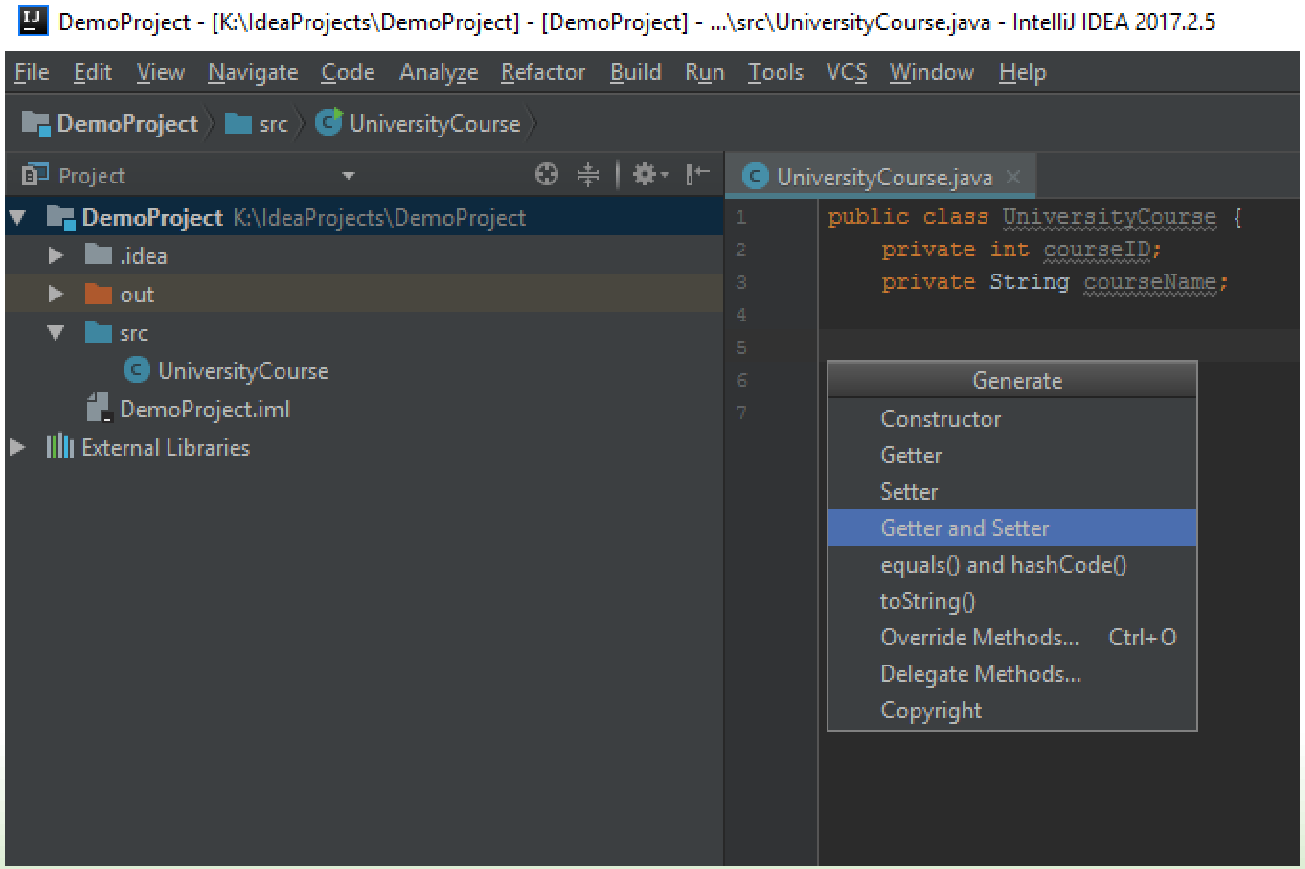
Getters and Setters are used to retrieve the values stored in the attributes of an instance of a class.

Getters and setters will usually be public access and allow interaction with attributes that should always be private access.

Add Getters and Setters

University of HUDDERSFIELD
Inspiring tomorrow's professionals

Then select the Getter and Setter option.



```
public class UniversityCourse {
    private int courseID;
    private String courseName;
```

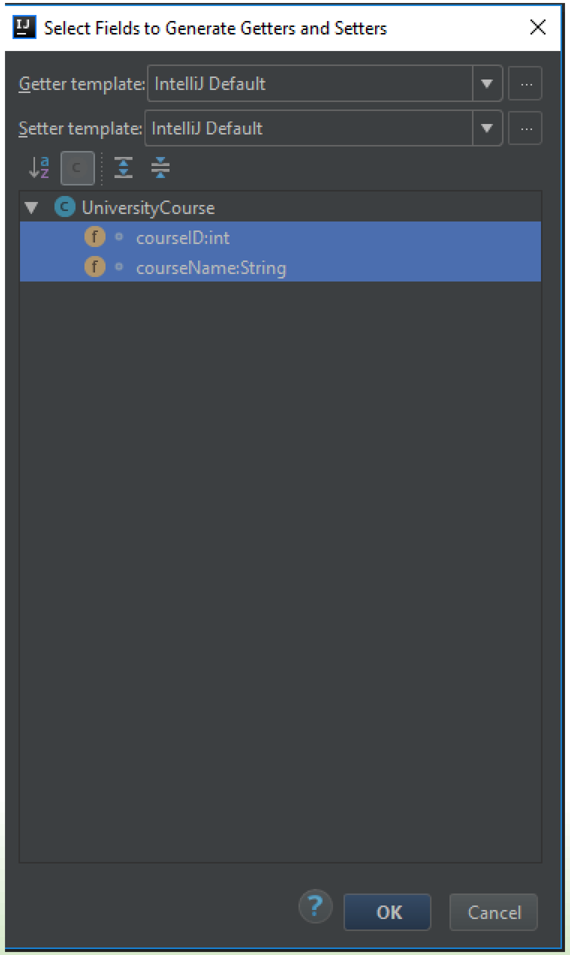
DemoProject - [K:\IdeaProjects\DemoProject] - [DemoProject] - ...\src\UniversityCourse.java - IntelliJ IDEA 2017.2.5
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
DemoProject src UniversityCourse
Project
UniversityCourse.java
DemoProject K:\IdeaProjects\DemoProject
.idea
out
src
UniversityCourse
DemoProject.iml
External Libraries
Generate
Constructor
Getter
Setter
Getter and Setter
equals() and hashCode()
toString()
Override Methods... Ctrl+O
Delegate Methods...
Copyright

Tip, your cursor needs to be inside the brackets of the class to generate these items.

Add Getters and Setters

University of HUDDERSFIELD
Inspiring tomorrow's professionals

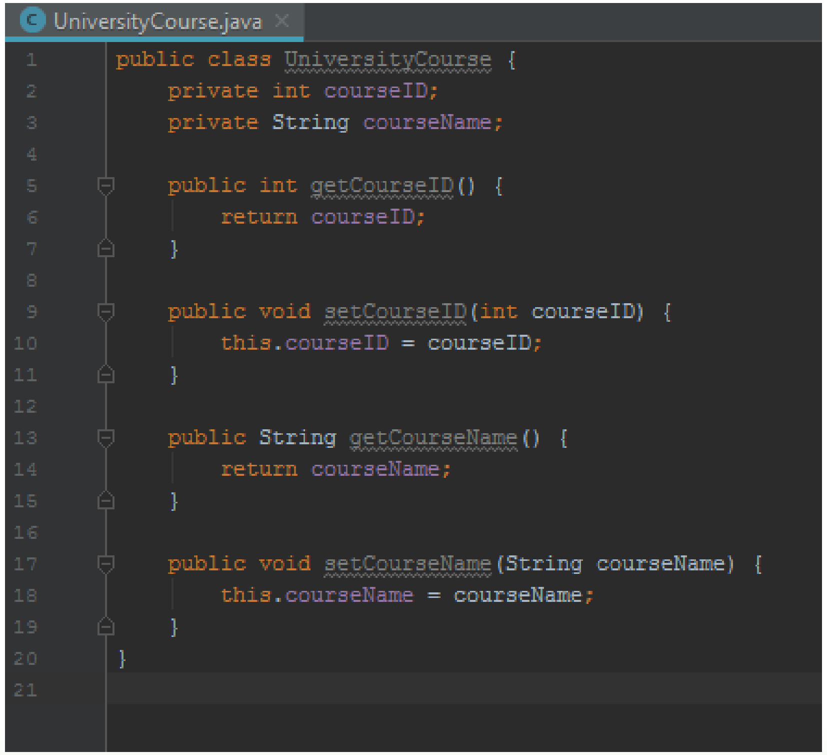
Highlight the attributes you wish to generate Getters and Setters for and click 'OK'



Select Fields to Generate Getters and Setters

Getter template: IntelliJ Default

Setter template: IntelliJ Default

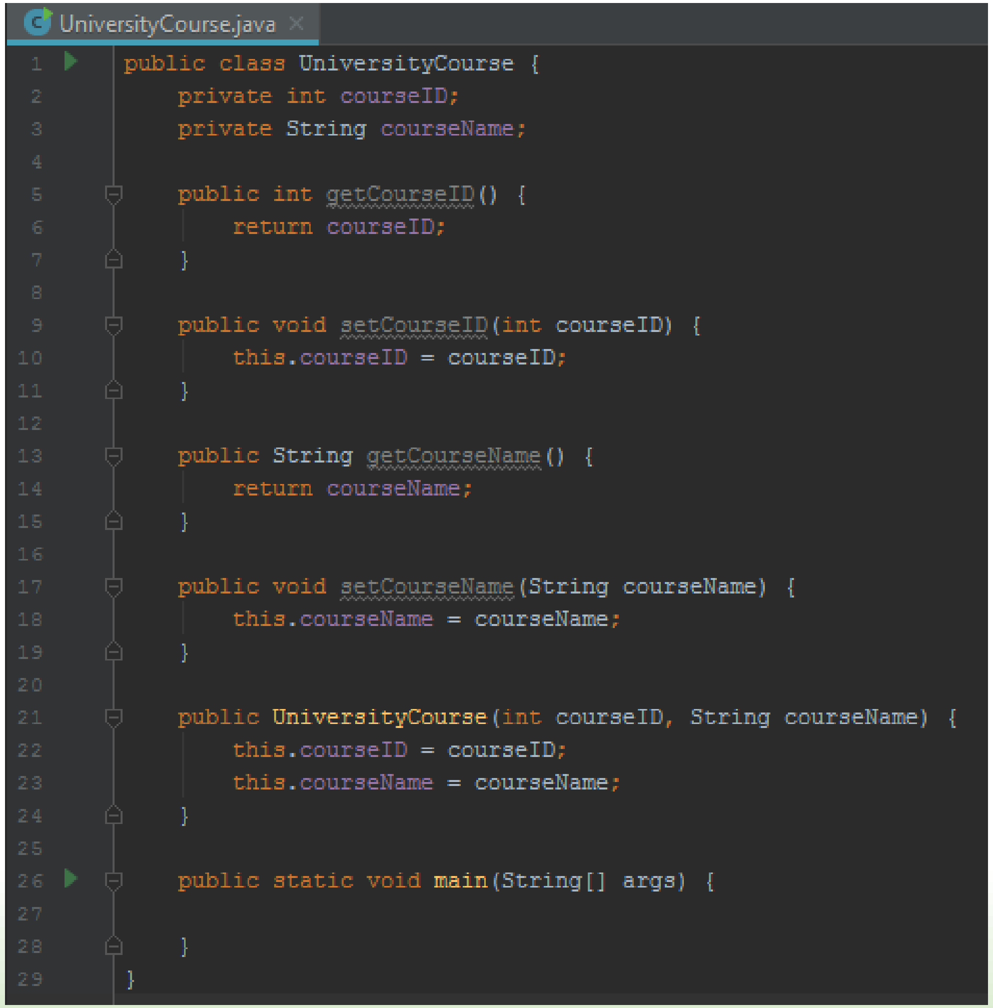


UniversityCourse.java

```java
public class UniversityCourse {
    private int courseID;
    private String courseName;

    public int getCourseID() {
        return courseID;
    }

    public void setCourseID(int courseID) {
        this.courseID = courseID;
    }

    public String getCourseName() {
        return courseName;
    }

    public void setCourseName(String courseName) {
        this.courseName = courseName;
    }
}

```

Your class should now look something like this, note the Set methods assign a value to the attribute. The Get methods return the value of the attribute.

Add the Rest



```java
public class UniversityCourse {
    private int courseID;
    private String courseName;

    public int getCourseID() {
        return courseID;
    }

    public void setCourseID(int courseID) {
        this.courseID = courseID;
    }

    public String getCourseName() {
        return courseName;
    }

    public void setCourseName(String courseName) {
        this.courseName = courseName;
    }

    public UniversityCourse(int courseID, String courseName) {
        this.courseID = courseID;
        this.courseName = courseName;
    }

    public static void main(String[] args) {

    }
}
```

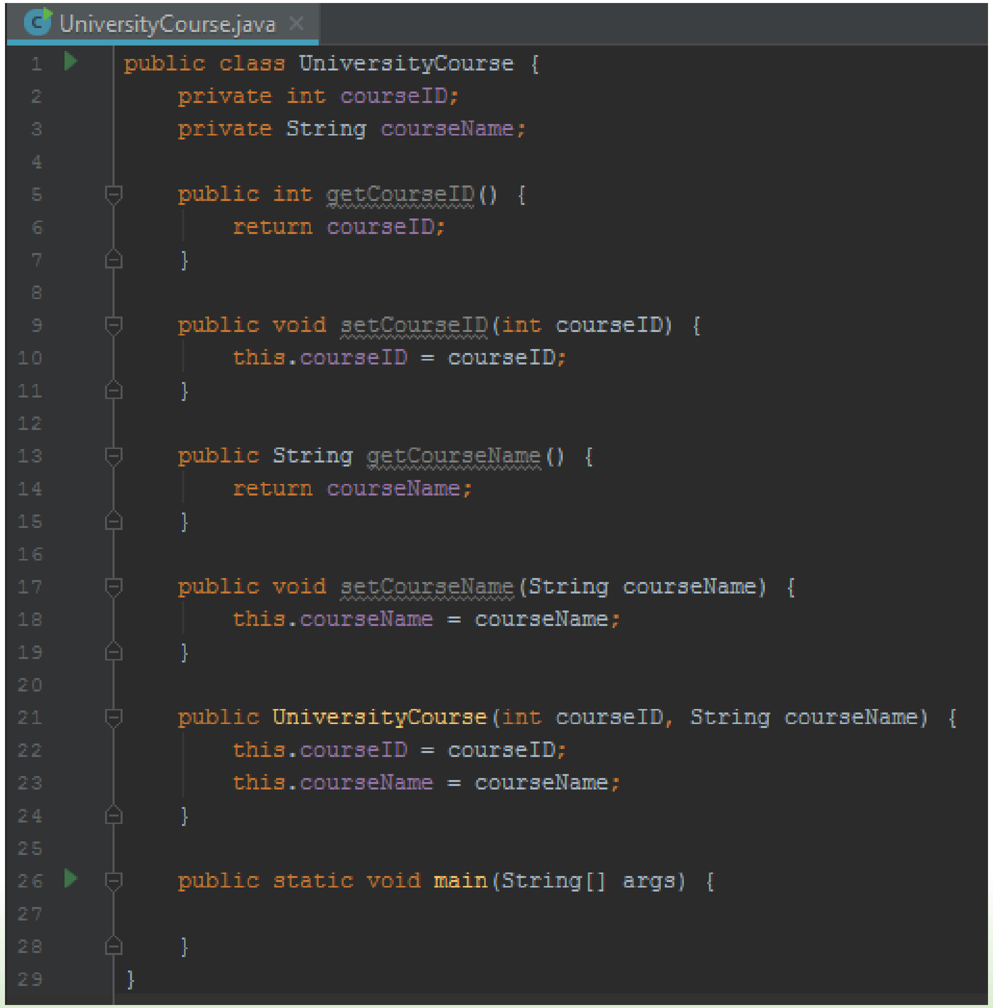
You can use the same process to add more class components, add the Constructor.

Type *psvm* into the class and IntelliJ's auto complete will suggest a method called Main(), add that too and you should have something like this.

Get into the habit of doing these actions EVERY time you create a Java class, once it becomes second nature it will save you a lot of time.

A Basic Class

University of
HUDDERSFIELD
Inspiring tomorrow's professionals



```java
public class UniversityCourse {
    private int courseID;
    private String courseName;

    public int getCourseID() {
        return courseID;
    }

    public void setCourseID(int courseID) {
        this.courseID = courseID;
    }

    public String getCourseName() {
        return courseName;
    }

    public void setCourseName(String courseName) {
        this.courseName = courseName;
    }

    public UniversityCourse(int courseID, String courseName) {
        this.courseID = courseID;
        this.courseName = courseName;
    }

    public static void main(String[] args) {

    }
}
```

We now have a basic (but lacking functionality) Java class.

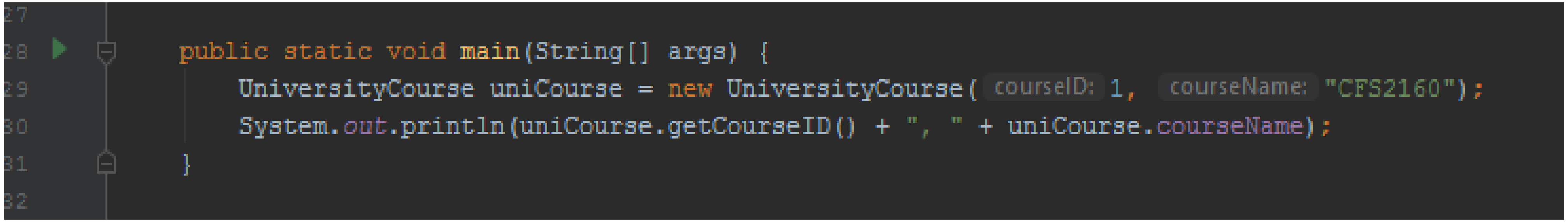
We need to add some more code to get it to do something.

A Basic Class

Write the following code into the Main method, pay attention to the suggestions made by IntelliJ, they ***can*** save you time. (IntelliJ can sometime offer the wrong suggestion so be careful!).

```
//create an instance of the class, pass the required values into the constructor
UniversityCourse uniCourse = new UniversityCourse(1, "CFS2160");
//use the get methods of the class to retrieve values and print them to the terminal
System.out.println(uniCourse.getCourseID() + ", " + uniCourse.getCourseName());
```

So it looks like this

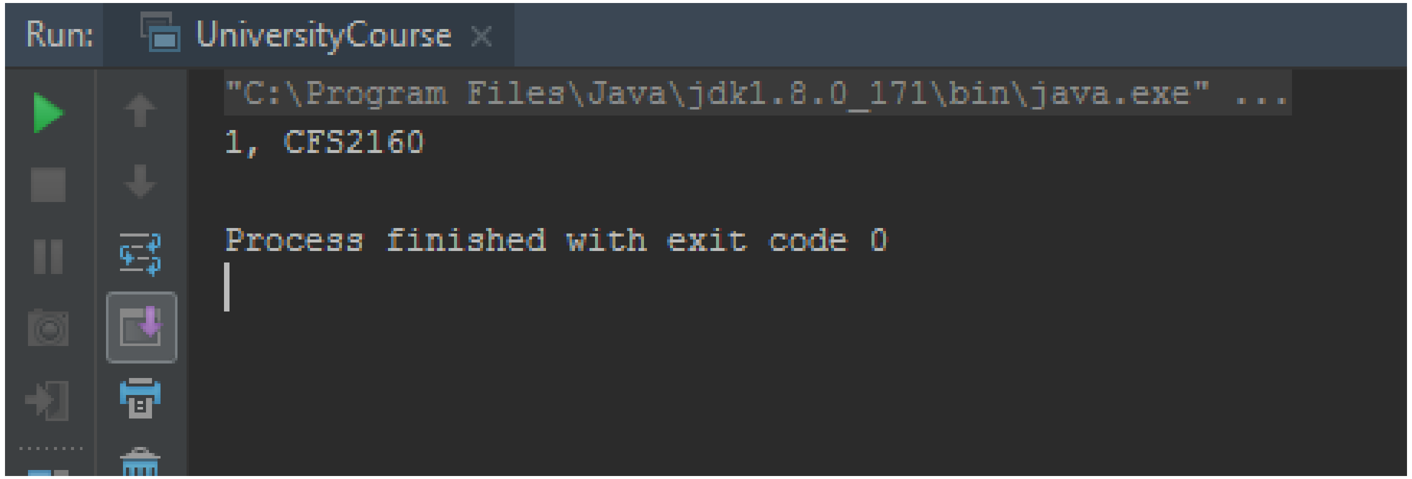


```
    public static void main(String[] args) {
        UniversityCourse uniCourse = new UniversityCourse( courseID: 1,  courseName: "CFS2160");
        System.out.println(uniCourse.getCourseID() + ", " + uniCourse.courseName);
    }
```

Finally click the green arrow and run the program.
The arrow only appears in class with a Main() method.

The Output

Just like in PyCharm, the output will be in the terminal window. Our program has outputted the values we added when we created a new object of the class UniversityCourse.



```
"C:\Program Files\Java\jdk1.8.0_171\bin\java.exe" ...
1, CFS2160

Process finished with exit code 0
```

Congratulations, you have just created an object of a Java class.
Change the values in the Main method where the constructor is called and see what happens.

Finally

From now on, your programming tutorials will be for java ONLY, if you have outstanding Python logbook work you should continue working on this in your own time. You can book 1 to 1 sessions with Tony or myself for help.

I will able to help with both Python and Java in my sessions.

- Any questions about Java?
- Look at todays worksheet