



OLLSCOIL NA GAILLIMHE  
UNIVERSITY OF GALWAY

**2023-2024 Semester 1**

**CT326 Programming III**

## Individual Assignment 2

<b>Objective</b>	Demonstrate the use of formatted strings, lambda expressions and anonymous inner classes; the use of an external jar; and Java programming fundamentals.
<b>Lecturer</b>	Dr Adrian Clear      Email: <a href="mailto:adrian.clear@universityofgalway.ie">adrian.clear@universityofgalway.ie</a>
<b>Marks Awarded</b>	This assignment is marked out of <b>60 marks</b> and is worth <b>~6%</b> of the overall grade for the module.
<b>Submission</b>	The deadline for submission is <b>23:59 on Wednesday, October 4<sup>th</sup>, 2023</b> .  Your attempt should be submitted through Canvas.  Please provide a <b>single zip file</b> of your answers. See further details below.

## Please complete all parts. All parts are worth equal marks.

1. Create an `Expense` class to represent an employee expense claim. An expense should have a date (`LocalDate`), a description (`String`), an expense category (a custom enum), and an amount (`joda.org.Money`). It should also have a `boolean` field to indicate whether it has been approved (false by default). You should define accessor methods for each of these, and the class should provide a means for approving expenses. This class should also contain a `toString` method for printing an expense in the format of the example below, using a *formatted* string.

Example string representation of an expense:

```
2022-09-20: Dell 17-inch monitor - EQUIPMENT - USD 540.00
```

See <https://www.joda.org/joda-money/> for details of using `joda Money` and related classes, examples of how to use this and related classes, and to download the jar.

2. Create an enum to represent the following expense categories:
  - Travel and subsistence
  - Supplies
  - Entertainment
  - Equipment
  - Other
3. Create an `ExpensesPortal` class that maintains a list of expenses and provides a means for submitting a new expense. It should also have a method called `printExpenses` with the following signature:

```
public void printExpenses(ExpensePrinter printer)
```

`ExpensePrinter` is an interface that you must define with a single method called `print` that takes a list of expenses as a parameter and prints the expenses to the console in a particular format.

4. Add a `main` method in the `ExpensesPortal` class or a new class. Create some expenses and add them to the an instance of `ExpensesPortal`.
5. Call the `printExpenses` method using a lambda expression to implement the `ExpensePrinter` parameter. The implementation should print a list of expenses to the console as follows:

Sample output:

```
2022-09-23: Flight to Glasgow - TRAVEL_AND_SUBSISTENCE - EUR 270.59
```

```
2022-09-20: Dell 17-inch monitor - EQUIPMENT - USD 540.00
```

```
2022-09-21: Java for Dummies - OTHER - EUR 17.99
```

6. Call the `printExpenses` method using an anonymous inner class to implement the `ExpensePrinter` parameter. The implementation should print a summary of expenses to the console as follows:

Sample output:

```
There are 3 expenses in the system totalling to a value of EUR 769.18.
```

Hint: create a static method in `ExpensesPortal` for summing the expenses. The method should support expenses in two currencies (EUR and USD). The return type should be of type `joda.org.Money` and should be in EUR, so you'll need to convert from USD to EUR to support USD expenses. See the [joda.org](http://joda.org) link above for examples of how to do this.

7. Finally, create a class called `PrinterByLabel` that implements the `ExpensePrinter` interface. It should organise the expenses by label before printing them out. Call the `printExpenses` method using a `PrinterByLabel` instance as a parameter.

Sample output:

*TRAVEL\_AND\_SUBSISTENCE*

*2022-09-23: Flight to Glasgow - TRAVEL\_AND\_SUBSISTENCE - EUR 270.59*

*SUPPLIES*

*ENTERTAINMENT*

*EQUIPMENT*

*2022-09-20: Dell 17-inch monitor - EQUIPMENT - USD 540.00*

*OTHER*

*2022-09-21: Java for Dummies - OTHER - EUR 17.99*

## Submission instructions:

Include your name and student number in a comment in each java source file and explain your code with comments. Submit your work in a single zip file using the name format

**assignment1\_firstname\_lastname.zip** via Canvas - other file formats will NOT be accepted, and email submission will NOT be accepted. Include a screen shot of the console output in the same zip file. Use proper indentation in your source code. If you submit more than one attempt, then only the last attempt will be marked.

**Plagiarism will be awarded a grade of zero!**