



# CFS2160: Programming Stream

## Tutorial/Practical 12

### Objects

#### Introduction

**There is no Log Book this term.**

The exercises in the lab will gradually build up a library of code that you will be able to draw on when you start work on the second assignment.

#### Activities

In the lecture one of the examples was part of a simple "To Do" list application.

Implement classes that could be used in such an application. You will need two classes:

1. Job. To represent a single job in the diary. It has a description, a priority (an integer) and a Boolean field to show whether or not it has been completed.
2. Diary. Which contains and manages a collection of jobs.

The precise design is up to you.

The application will need to print a list of all current jobs, sort jobs into order of priority (high numbers indicate a high priority), print only those jobs that contain a certain string, and to remove completed jobs. Use these requirements (along with the suggestions below) to decide on what methods to implement in the classes.

Once the classes are complete, write a program that:

1. Creates five jobs, with different priorities, none of which are complete.
2. Adds all these jobs to a diary.
3. Prints the diary.
4. Prints only the jobs that contain a certain string.
5. Sorts the diary by priority.
6. Prints the diary.
7. Marks one job as complete.
8. Removes complete jobs from the diary.
9. Prints the diary.

## Suggestions

Job is a simple class with the three attributes, constructors, getters and setters and probably a method to print the job details neatly. A simple method to mark a job as completed will make things a little neater. The method signatures might be:

```
public Job ()
public Job (String description, int priority)
public String getDescription ()
public void setDescription (String description)
public int getPriority ()
public void setPriority (int priority)
public boolean isFinished ()
public void setFinished (boolean finished)
public void markFinished ()
public int compareTo (Job other)
public String display ()
```

Diary will need methods to add a job, display all the current jobs, display the jobs where the description contains some string, sort the current jobs, and remove completed jobs. The method signatures could be:

```
public Diary ()
public void addJob (Job j)
public void printDiary ()
public void searchDiary (String search)
public void purgeDiary ()
public void sortDiary ()
```