COMP41100

EXPLORING PROGRAMMING IN RUBY

PRACTICAL FIVE

24th October 2013

Andrew Doyle
Student Number: 12252388

University College Dublin

PART ONE

Task:

Re-Visit the iTunes program and do the v.3 of it. Don`t try to cut and paste what I have given you, work through the description I give of its properties and try to appreciate why these changes are being proposed and, if possible, improve on them yourself.

Solution:

The author attempted to implement v.3 of the programme by altering the v.2 provided. The author attempted the removal of class variables and altering methods to attempt to run the programme without error. The attempts to date are included in the submission.

PART TWO

Task:

Define three classes, calling them something much more original than A, B and C; set them up so that C is a subclass of B and B is a subclass of A. Then define a bunch of attributes in A and about 3 methods and test their inheritance to instances of C. Define at least one additional method for B and C respectively and note how they do not apply up the hierarchy but they do apply down (if there is indeed a down).

Solution:

- Class “A” is ClubDepartments
- Class “B” is FootballPlayers
- Class “C” is FootballCoaches

The three methods defined in Class “A” (ClubDepartments) are:

1. **yearly_cost**
2. **monthly_cost**
3. **weekly_cost**

Any of the above methods can be called on any instance of the three classes, since the classes are extended using the extends symbol: **<**

The method defined in Class “B” (FootballPlayers) is:

age_b

The above method can be called on either the **FootballPlayers** class or the **FootballCoaches** class. However it cannot be called on an instance of the **ClubDepartments** class. This shows how the method applies down the hierarchy of classes, but not up the hierarchy.

The method defined in Class “C” (FootballCoaches) is:

age_c

The above method can **only** be called on an instance of the **FootballCoaches** class. This demonstrates how the method does not apply up the hierarchy.