

University of Hull
Department of Computer Science

Wrestling with Python – Week 3
Methods and Testing

Vsn. 1.0 Rob Miles 2014

Introduction

Welcome to our Python sessions.

Please follow the instructions carefully. You need to enter your programs exactly as they are written, otherwise they will do the wrong thing, or nothing at all.

In this session you will explore the use of methods in your programs.



This indicates an activity you should perform in at this point in the text. You may be given precise instructions, or you may have to work something out for yourself.

The Degree Classifier

I want a method that will accept a student score and then return a string that gives the degree classification that the mark represents. The rules are:

- Any mark less than 40 is a fail
- Any mark between 40 and 49 is a Third
- Any mark between 50 and 59 is a 2(ii)
- Any mark between 60 and 69 is a 2(i)
- Any mark greater than 69 is a First

We have hired a Python programmer and they have come up with the following:

```
def degree(mark):  
    if mark <= 40:  
        return "Fail"  
    if mark <= 50:  
        return "Third"  
    if mark <=60:  
        return "2(ii)"  
    if mark <= 70:  
        return "2(i)"  
    return "First"  
  
print(degree(55))
```

The programmer claims it works because when they run it the program prints “2(ii)” which is the correct score.



Before you go any further; perform the following:

1. Start PyCharm.
2. Open a new editing window.
3. Enter the code above and run it. Make sure it prints the correct classification.

Testing the Degree Classifier

Now you need to think up some numbers that will give the program a proper test. Try the program with numbers around the boundaries (for example 59, 60 and 61) and see if what happens agrees with the specification.

Black and White Box Testing

In programming terms, making calls of the method and seeing what it does is an example of “black box” testing. The method is regarded as “black box” which hides the code inside and the testing is performed by sending values into the method and looking at what comes out.

White Box testing is when you look at the code itself and see if it makes sense. Consider this condition from the `degree` method:

```
if mark <= 40:  
    return "Fail"
```

If the mark is 40 the condition is true and the method will return “Fail”, which is wrong according to the spec. We don’t actually need to run any code through the program to test like this, we can just see instantly what is wrong.



1. Take a look at the code from a White Box testing point of view and see if you can spot the mistakes.

When it is important that a program works correctly it is useful to use a mix of black and white box testing. The white box testing process is sometimes called a “code review” where one programmer presents their code to others who go through each line and convince themselves that it will work.

Creating and Testing a ReadNumber method

It might be useful to create a number fetching method that will display a prompt message and perform validation to make sure that the value supplied is not too large or too small. This is how it could be used to select the number of a film the user wants to see, in the range 1 to 6:

```
FilmNo = ReadNumber ("Enter film number : ", 1, 6)
```

The method will repeatedly read numbers from the user of the program until they enter a value which is in the range 1-6. We could use a slightly different call to get the age of the person going to see a film:

```
Age = ReadNumber ("Enter your age : ", 3, 100)
```

This time the method would display a different prompt and repeatedly ask for values until one was returned in the range 3-100.

The method has three parameters, that we give names in the method definition:

```
def ReadNumber (prompt, lowValue, highValue):  
    return 1
```

This is not the completed method, it returns 1 every time it is called, but this should show you what the method looks like.



1. Consider how you would test a ReadNumber method. What are the ways that the method could fail?

Rob Miles

November 2014