T-106.1208 Basics of Programming Y (Python). Exam 8.5.2009

Write the following information clearly on top of each paper you submit: name of the course, date of the exam, your full name, student ID, the total number of papers you submit, and your signature.

Important instructions: Use indentations of the length of two squares in your code. If your indentations are not clear enough, you lose points. You do not have to write any comments in your code. You can assume that the input given by the user is correct, if it is not told in the problem that you should handle the incorrect input.

1. a) What is printed when the following Python program is executed? It is enough to give the correct output without any explanations. (2 p)

```
def main():
    overspeed = 25.0
    if overspeed > 10.0:
        print "Fixed penalty"
    elif overspeed > 20.0:
        print "Day fine"
    else:
        print "No penalty"


main()
```

b) Suggest an initial value for each of the variables age and day that will cause the following Python program print `Full price`. It is enough to give one possible value for each variable. You do not have to list all possibilities. (2 p)

```
def main():
    age = ...  # Three dots are replaced with an integer
    day = ... # Three dots are replaced with a string
    if age <= 6:
        print "Free entry"
    else:
        if day == "Sat" or day == "Sun":
            print "Free entry"
        else:
            print "Full price"

main()
```

c) What is printed when the following Python program is executed? It is enough to give the correct output without any explanations. (3 p)

```
def main():
    numbers = [1, 25, 4, 9, 16]
    result = 100
    for number in numbers:
        result = result - number
    print result


main()
```

d) Explain, in plain English using 1-2 sentences, the *purpose* of the following Python function **mystery1**. (Do not say *how* the code works. Instead say what the function would be used for.) You may assume that the parameters of function are two equally-long lists of integers. (4 p)

```
def mystery1(numbers1, numbers2):
    result = 0
    i = 0
    while i < len(numbers1):
        if numbers1[i] != numbers2[i]:
            result = result + 1
        i = i + 1
    return result
```

e) Explain, in plain English using 1-2 sentences, the *purpose* of the following Python function **mystery2**. (Do not say *how* the code works. Instead say what the function would be used for.) You may assume that the first parameter of the function is a list containing decimal numbers (floats) and the second parameter is a decimal number. (4 p)

```
def mystery2(numberlist, number):
    result = 0.0
    for element in numberlist:
        result = result + number * element
    return result
```

f) Explain, in plain English using 1-2 sentences, the *purpose* of the following Python function **mystery3**. (Do not say *how* the code works. Instead say what the function would be used for.) You may assume that the parameter of the function is a list containing integers. (5 p)

```
def mystery3(numbers):
    i = 0
    while i < len(numbers) - 1:
        if numbers[i] <= numbers[i + 1]:
            return False
        i = i + 1
    return True
```

g) Explain, in plain English using 1-2 sentences, the *purpose* of the following Python function `mystery4`. (Do not say *how* the code works. Instead say what the function would be used for.) You may assume that the first parameter of the function is a string and the second parameter is an integer. (5 p)

```
def mystery4(text, number):
    if len(text) >= number:
        return text[0:number]
    else:
        return text + (" " * (number - len(text)))
```

2. a) A shop gives reductions of the prices as follows: If the original price of an item is at least 200 euros, the reduction is 10 per cent. If the original price is at least 100 euros, but less than 200 euros, the reduction is 5 per cent. Write a Python program which asks the user to give an original price of an item and outputs the reduced price.(10 p.)

   b) A certain course has a following system: There are several exercise rounds in the course. All rounds have the same minimum limit. If a student's points are less than the minimum limit in any roud, the student has failed the exercises. If the student has at least minimum number of points in all exercise rounds, he/she has passed the exercises. Write a Python function `exercises_passed(points, minimum)`. The first parameter of the function is the list containing the student's exercise points from the exercise rounds (for example, the first element of the list is the total exercise points from round 1, the second element is exercise points from round 2 and so on. The points are integers) and the second parameter is an integer which represents the minimimum limit. The function returns value `True`, if the student has passed the exercises and value `False`, if the student has failed the exercises. Write only this function, the main program or other functions are not required. (20 p)

3. Write a program which asks the user to input a file name. The program reads this file, which contains temperatures (decimal numbers) such that each temperature is alone in its own row. The number of the temperatures is not known beforehand. After the program has read the file, it should output the number of temperatures under 0 degrees, the number of temperatures between 0 - 25 degrees and the number of temperatures over 25 degrees. The example of the output is given below:

```
Temperature under 0 degrees 4 times.
Temperature 0 - 25 degrees 10 times.
Temperature over 25 degrees 2 times.
```

   The program must be able to handle the following errors

   - The file does not exists or it is not possible to read the file because of some other reason.
   - The file contains a defective row (for example, the row contains other text in addition to a decimal number or there is no number at all in the row).

   In these cases, the program outputs an error message and stops. The program does not have to continue reading after the defective row or output the distribution of the temperatures it has read before the error. (20 p.) **CONTINUES**

If you find this problem two difficult, you may write the program which does not satisfy all requirements. For example, you may write the program which does not handle the errors or you may write the program which reads the input from the user instead of the file. However, this will reduce your points. If you write a program which does not handle errors, the maximum number of points from this problem is 15. If you write a program which reads input from the user instead of the file, the maximum is 7 points.

4. A firm has a game in its web page. The user can play the game alone (against the computer) and he/she is given points according to the result (the higher points the better). The firm wants to maintain statistics of the games played. Write a class `Player` to describe one player (you are not asked to write any other parts of the program except this class and the main function described in the end).

The class `Player` has the following data attributes:

- `__name` name of the player
- `__number_of_games` the number of the games this player has played
- `__record` the record of this player (the highest points he/she has achieved in his/her games)
- `__total_points` the total number of points the player has achieved in all games he/she has played.

Write the following methods in your class. (To shorten your solution, some methods which would be useful have been omitted. If the description does not say anything about the return value of a method, the method does not have to return anything.)

- `__init__(self, player_name)` creates a new `Player` object. The name of the player is given as the parameter. The record, number of games and total points of the new player are 0.
- `get_record(self)` returns the record of this player.
- `add_game(self, points)` adds information of the new game result to this player. The points of the game are given as a parameter. The method updates the number of games and the total points and also the record if necessary.
- `calculate_average(self)` calculates and returns the average number of points this player has obtained from all his/her games.
- `master_player(self)` returns value `True` if this player is a master player and `False` otherwise. The player is a master player if his/her record is at least 4500 points and he/she has played at least 50 games (the both conditions must be true). Otherwise, he/she is not a master player.
- `__str__(self)` returns a string which contains the name, record and number of games of this player.

In addition, write a main function which creates two `Player` objects, calls two times `add_game` method for both players and after that once `get_record` and `calculate_average` methods for the second player. The program must also output the record and the average returned by the functions. Next, the program must find out if the second player is a master player and output either "The player is a master player" or "The player is not a master player" accordingly. Finally, the program has to output basic information (the name, the record and the number of games) about both players. You may choose the names of the players and the results of the games yourself. Your program does not have to ask any input from the user. (25 p)