Appendix E: Project Answers

Exercise 1

```
from random import randint
from os import remove, rename
```

Exercise 2

```
def getUserScore(userName):

    try:
        input = open('userScores.txt', 'r')
        for line in input:
            content = line.split(',')
            if content[0] == userName:
                input.close()
                return content[1]

        input.close()
        return "-1"
    except IOError:
        print ("\nFile userScores.txt not found. A new file will be
created.")
        input = open('userScores.txt', 'w')
        input.close()
        return "-1"
```

Exercise 3

```
def updateUserPoints(newUser, userName, score):

    if newUser:

        input = open('userScores.txt', 'a')
        input.write(userName + ', ' + score)
        input.close()
    else:

        input = open('userScores.txt', 'r')
        output = open('userScores.tmp', 'w')

        for line in input:
```

```
            content = line.split(',')
            if content[0] == userName:
                content[1] = score
                line = content[0] + ', ' + content[1] + '\n'

            output.write(line)

        input.close()
        output.close()

        remove('userScores.txt')
        rename('userScores.tmp', 'userScores.txt')
```

Exercise 4

-

```
def generateQuestion():

    operandList = [0, 0, 0, 0, 0]
    operatorList = ['', '', '', '']
    operatorDict = {1:' + ', 2:' - ', 3:'*', 4:'**'}

    for index in range(0, 5):
        operandList[index] = randint(1, 9)

    for index in range(0, 4):
        if index > 0 and operatorList[index-1] != '**':
            operator = operatorDict[randint(1, 4)]
        else:
            operator = operatorDict[randint(1, 3)]

        operatorList[index] = operator

    questionString = str(operandList[0])

    for index in range(1, 5):
        questionString = questionString + operatorList[index-1] +
str(operandList[index])

    result = eval(questionString)

    questionString = questionString.replace("**", "^")

    print ('\n' + questionString)
```

```
    userResult = input('Answer: ')

    while True:
        try:
            if int(userResult) == result:
                print ("So Smart")
                return 1
            else:
                print ("Sorry, wrong answer. The correct answer is",
result)
                return 0
        except Exception as e:
            print ("You did not enter a number. Please try again.")
            userResult = input('Answer: ')
```

[Explanation for Exercise 4.2]

Starting from the second item (i.e. index = 1) in `operatorList`, the line `if index > 0 and operatorList[index-1] != '**':` checks if the previous item in `operatorList` is the '**' symbol..

If it is not, the statement `operator = operatorDict[randint(1, 4)]` will execute. Since the range given to the `randint` function is 1 to 4, the numbers 1, 2, 3 or 4 will be generated. Hence, the symbols '+', '-', '*' or '**' will be assigned to the variable `operator`.

However, if the previous symbol is '**', the else statement (`operator = operatorDict[randint(1, 3)]`) will execute. In this case, the range given to the `randint` function is from 1 to 3. Hence, the '**' symbol, which has a key of 4 in `operatorDict` will NOT be assigned to the `operator` variable.

Exercise 5

```
try:

    import myPythonFunctions as m

    userName = input('''Please enter your user name or
create a new one if this is the first time
```

```
you are running the program: ''')

    userScore = int(m.getUserScore(userName))

    if userScore == -1:
        newUser = True
        userScore = 0
    else:
        newUser = False

    userChoice = 0

    while userChoice != '-1':

        userScore += m.generateQuestion()
        print ("Current Score = ", userScore)
        userChoice = input("Press Enter To Continue or -1 to Exit: ")

    m.updateUserPoints(newUser, userName, str(userScore))

except Exception as e:
    print ("An unexpected error occurred. Program will be exited.")
```

Challenge Yourself

You only need to change the function `generateQuestion()` for all the challenges. Here's the suggested solution.

```
def generateQuestion():
     operandList = [0, 0, 0, 0, 0]
     operatorList = ['', '', '', '']
     operatorDict = {1:' + ', 2:' - ', 3:'*', 4:'/', 5:'**'}

     result = 500001

     while result > 50000 or result < -50000:
          for index in range(0, 5):
               operandList[index] = randint(1, 9)

          for index in range(0, 4):
               if index > 0 and operatorList[index-1] != '**':
                    operator = operatorDict[randint(1, 5)]
               else:
```

```
                    operator = operatorDict[randint(1, 4)]

               operatorList[index] = operator

          '''
          Randomly generate the positions of ( and )
          E.g. If openBracket = 2, the ( symbol will be placed in
          front of the third number
          If closeBracket = 3, the ) symbol will be placed behind the
          fourth number
          Since the closing bracket cannot be before the opening
          bracket,we have to generate the position for the closing
          bracket from openBracket + 1 onwards
          '''


          openBracket = randint(0, 3)
          closeBracket = randint(openBracket+1, 4)

          if openBracket == 0:
               questionString = '(' + str(operandList[0])
          else:
               questionString = str(operandList[0])

          for index in range(1, 5):
               if index == openBracket:
                    questionString = questionString +
operatorList[index-1] + '(' + str(operandList[index])
               elif index == closeBracket:
                    questionString = questionString +
operatorList[index-1] + str(operandList[index]) + ')'
               else:
                    questionString = questionString +
operatorList[index-1] + str(operandList[index])

          result = round(eval(questionString), 2)

          #End of While Loop

     questionString = questionString.replace("**", "^")

     print ('\n' + questionString)
```

```
     userResult = input('Answer (correct to 2 d.p. if not an
integer): ')

     while True:
          try:
               if float(userResult) == result:
                    print ("So Smart")
                    return 1
               else:
                    print ("Sorry, wrong answer. The correct answer
is", result)
                    return 0
          except Exception as e:
               print ("You did not enter a number. Please try
again.")
               userResult = input('Answer (correct to 2 d.p. if not
an integer): ')
```