

Teaching guide: Programming challenge 1

Programming is a fundamental skill required for success in GCSE Computer Science. This programming challenge is designed to develop students' programming skills.

In the paper 1 exam, students will be required to design, write, test and refine program code in either C#, Python (version 3) or VB.Net.

To develop their programming skills, successful students should have sufficient practical experience of:

- structuring programs into modular parts with clear documented interfaces to enable them to design appropriate modular structures for solutions
- including authentication and data validation systems/routines within their computer programs
- writing, debugging and testing programs to enable them to develop the skills to articulate how programs work and argue using logical reasoning for the correctness of programs in solving specified problems
- designing and applying test data (normal, boundary and erroneous) to the testing of programs so that they are familiar with these test data types and the purpose of testing
- refining programs in response to testing outcomes.

Programming challenge 1 - User registration

Exercise 1

This exercise can be solved very simply, with a nested IF statement or a flag to indicate validation. **Figure 1** shows an example solution written in Python.

Specification coverage: sections 3.2.2, 3.2.4, 3.2.6, 3.2.7 and 3.2.11

Figure 1

```
#####  
# AQA User Registration #  
#####  
  
userNames = ["User1", "User2", "User3", "User4", "User5"]  
  
print("Welcome to AQA User Registration")  
  
userName = input("New username: ")  
  
if userName in userNames:  
    print("The username is already used")  
else:  
    userPassword = input("Password (12+ chars or more): ")  
  
    if len(userPassword) < 12:  
        print("The password is not long enough")  
    else:  
        print("Your username is " + userName)  
        print("Your password is " + userPassword)  
  
# end of program
```

Exercise 2

There are 4 errors in **Figure 2**.

- Invalid syntax line 13, colon missing from the else statement.
- Indented block error line 19, print statements should be indented in the else statement.
- Logic error line 16, validation for password length only checks for 12 characters, should be less than 12 (<12).
- Incorrect spelling of variable name in line 16 (userpassword).

Specification coverage: section 3.2.11

Figure 2

```
#####  
# AQA User Registration #  
#####  
  
userNames = ["User1", "User2", "User3", "User4", "User5"]  
  
print("Welcome to AQA User Registration")  
  
userName = input("New username: ")  
  
if userName in userNames:  
    print("The username is already used")  
else:  
    userPassword = input("Password (12+ chars or more): ")  
  
    if len(userpassword) == 12:  
        print("The password is not long enough")  
    else:  
        print("Your username is " + userName)  
        print("Your password is " + userPassword)  
  
# end of program
```

Exercise 3

This exercise requires students to add more complex code. Although the specification does not contain file handling, it can be used to demonstrate data structures in some context. Alternatively, students can define a data structure that can be added to.

The solution allows students to use nested selection and iteration. **Figure 3** is an example solution written in Python using files but the lists can be populated in the code instead.

Specification coverage: section 3.2.2

Figure 3

```
#####  
# AQA User Registration #  
#####  
  
# define a list and open a file called usernames.txt  
# read the text file and add it to the list  
  
myFile = open("usernames.txt", "r")  
userNames = myFile.read().splitlines()  
myFile.close()  
  
# A new list is created with usernames in lower case  
# for validation that is not case sensitive  
userNamesLower = [un.lower() for un in userNames]  
  
# welcome and input interface  
print("Welcome to AQA User Registration")  
  
tries = 0 # only 3 tries are allowed  
validUserName = False  
  
# loop up to 3 times for a new username  
while (tries < 3) and (not validUserName):  
    userName = input("New user name: ")  
    if userName.lower() in userNamesLower:  
        print("The username is already used")  
        tries += 1  
    else:  
        validUserName = True
```

```
if validUserName:
    tries = 0
    validPassword = False
    while (tries < 3) and (not validPassword):
        userPassword = input("Password (12+ chars): ")
        if len(userPassword) < 12:
            print("The password is not long enough")
            tries += 1
        else:
            print("Your username is " + userName)
            print("Your password is " + userPassword)
            validPassword = True

            # add the new username to the text file
            userNames.append(userName)
            # open the file for appending (the new name)
            myFile = open("usernames.txt", "a")
            myFile.write(userName + '\n')
            myFile.close()

# error message if the user has failed validation
if tries == 3:
    print("Only three tries are allowed.")
else:
    print("Thank you for registering.")

# end of program
```

Exercise 4

This exercise adds a menu, which is a common requirement for programs. It also asks students to use subroutines.

Figure 4 shows a simple example solution, using Python code from earlier exercises.

More sophisticated solutions could use csv files, and more parameter passing rather than using global variables.

Specification coverage: section 3.2.10

Figure 4

```
#####  
# AQA User Registration #  
#####  
  
def DisplayMenu():  
    print("")  
    print("#####")  
    print("Please enter a menu choice")  
    print("Enter 1 to login with username & password")  
    print("Enter 2 to add a new username & password")  
    print("Enter 3 to quit the program")  
    print("#####")  
  
def UserDetails():  
    # define a list and open file called usernames.txt  
    # read the text file and add it to the list  
    # repeat for passwords file. Each username and  
    # password combination is on the same line in  
    # each file.  
  
    myFile = open("usernames.txt", "r")  
    userNames = myFile.read().splitlines()  
    myFile.close()
```

```
myFile = open("passwords.txt", "r")
passwords = myFile.read().splitlines()
myFile.close()

# A list is created with usernames in lower case
# for validation that is not case sensitive
userNamesLower = [un.lower() for un in userNames]

return (userNames, passwords, userNamesLower)

def RegisterNew(userNamesLower):
    tries = 0 # only 3 tries are allowed
    validUserName = False # set if username is valid

    # loop up to 3 times for a new username
    while (tries < 3) and (not validUserName):
        uName = input("New username: ")
        if uName.lower() in userNamesLower:
            print("The username is already used")
            tries += 1
        else:
            validUserName = True

    if validUserName:
        tries = 0
        validPassword = False
        while (tries < 3) and (not validPassword):
            uPassword = input("Password (12+ chars): ")
            if len(uPassword) < 12:
                print("The password not long enough")
                tries += 1
            else:
                print("Username is " + uName)
                print("Password is " + uPassword)
                validPassword = True

        # add the new username to the text file
        userNames.append(uName)
        myFile = open("usernames.txt", "a")
        myFile.write(uName + '\n')
        myFile.close()

        myFile = open("passwords.txt", "a")
        myFile.write(uPassword + '\n')
        myFile.close()
```

```
# error message if the user has failed validation
if tries == 3:
    print("Only three tries are allowed")
else:
    print("Thank you for registering.")

def Login(un, pw, userNamesLower, passwords):
    # checks username is found and the password matches
    if un.lower() in userNamesLower:
        i = userNamesLower.index(un.lower())
        if pw == passwords[i]:
            return True
        else:
            return False
    else:
        return False

# Main program

userNames, passwords, userNamesLower = UserDetails()
print("Welcome to AQA User Registration")

#loop that allows a menu number to be entered
userInput = ''
while userInput != "3":
    DisplayMenu()
    userInput = input("Menu choice: ")

    if userInput == "1":
        un = input("Please enter your username: ")
        pw = input("Please enter your password: ")
        print("")
        loginOK = Login(un, pw, userNamesLower, passwords)
        if loginOK:
            print("Login successful")
        else:
            print("Login incorrect: register or retry")
    elif userInput == "2":
        RegisterNew(userNamesLower)
    elif userInput == "3":
        print("Goodbye")
    else:
        print("ERROR! you must enter 1, 2 or 3")

# end of program
```