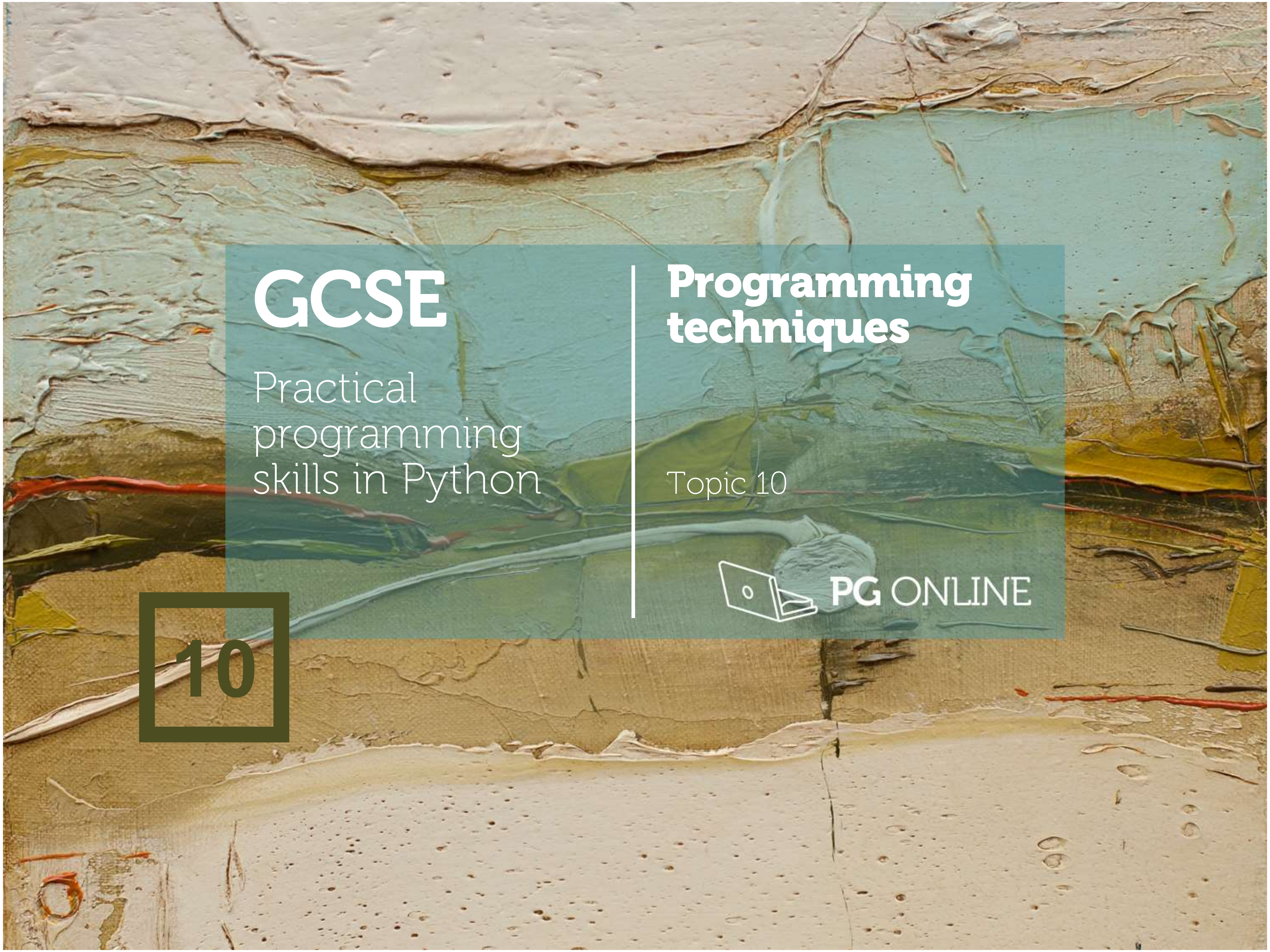

GCSE

Practical programming skills in Python

Programming techniques

Topic 10

10

PG ONLINE

Objectives

- Use programming ’set pieces’ for common problems
  - Use a ‘flag’
  - Use a while loop to validate data entry
  - Create a menu system

Starter Activity

- Which of these lists are in ascending order?

```
listA = [12, 17, 42, 56, 68]
listB = [21, 23, 39, 28, 42]
listC = [-12, -3.7, 2.01, 2.23]
```

Starter Activity

- Which of these lists are in ascending order?

```
listA = [12, 17, 42, 56, 68]          √
listB = [21, 23, 39, 28, 42]          X
listC = [-12, -3.7, 2.01, 2.23]       √
```

Flag

- A ’flag’ is a variable that is used to check if something has happened
  - A flag is usually a binary, or Boolean value, e.g. True or False

Flag

- This program will check if an answer is correct
- The name of the flag in this example is “correct”

```
correct = False
answer = int(input("12 x 3? "))
if answer == 36:
      correct = True
```

Flag

- This flag can be checked later

```
if correct == True:
    print("User was correct")
else:
    print("User was wrong")
```

Worksheet 10a

- Complete **Questions 1, 2** and **3**

The break statement

- This program checks whether a list is sorted in ascending sequence

```
myList = [12, 17, 42, 56, 68]
for i in range(len(myList)-1):
    if myList[i] < myList[i+1]:
        print("OK so far")
    else:
        print("Wrong order")
        break
```

- The `break` statement causes control to pass to the next statement after the `for` loop

Writing efficient code

- Printing out the result for each step is inefficient

```
myList = [12, 17, 42, 56, 68]
for i in range(len(myList)-1):
    if myList[i] < myList[i+1]:
        print("OK so far")
    else:
        print("Wrong order")
        break
```

- Can you rewrite the loop more efficiently?

Writing efficient code

- Rewrite the `if` statement:

```
valid = True # assume list is valid
myList = [12, 17, 42, 56, 68]
for i in range(len(myList)-1):
   if myList[i] > myList[i+1]:
      print("Wrong order")
      valid = False
      break
```

Using a flag and a while loop

- Using the `break` statement is not considered good practice.
- Instead, you can use a flag and a `while` loop

```
valid = True # assume list is valid
myList = [12, 17, 42, 56, 68]
i = 0
while valid == True and i < len(myList)-1:
(complete the code to print either
"List is unsorted" or
"List is in ascending order")
```

Using a flag

```python
valid = True # assume list is valid
myList = [12, 17, 42, 56, 68]
i = 0
while valid == True and i < len(myList)-1:
    if myList[i] > myList[i+1]:
        valid = False
        print("List is unsorted")
    else:
        i = i + 1
if valid == True:
        print("List is in ascending order")
```

Validation

- Try the following code:

```
choice = ""
while choice != "y" and choice != "n":
   choice = input("Enter ‘y’ or ‘n’: ")

if choice == "y" :
   print("You chose ‘yes’")
else:
   print("You chose ‘no’")
```

Validation

- Validating an input means checking that the data you have collected is reasonable or possible:
- Shoe shops generally sell shoes between sizes 1-13
- Create a program that will:
  - ask the user for a shoe size
  - give an error if the number entered is invalid
  - allow the user to try again until they enter a valid number

Validation

- Shoe size example:

```
size = 0
while size < 1 or size > 13:
    size = int(input("Enter a shoe size \
between 1 and 13: "))

print("Accepted")
```

Validation

- Alternative shoe size example:

```
size = int(input("Enter a shoe size:"))
while size < 1 or size > 13:
    size = int(input("Error, must be 1-13: "))

print("Accepted")
```

Validation

- Common rules for validation:
  - set an empty value or ask for a value first
  - create a while loop that will repeat while the answer is invalid
  - ask for a value **inside** the loop as well

Worksheet 10b

- Complete **Worksheet 10b**

Menu system

- Many programs would benefit from a menu system
- Using subroutines (functions and procedures) and validation together makes this really easy

Menu system

- First, create some empty procedures:

```python
def option1():
    print("Option 1 run")

def option2():
    print("Option 2 run")
```

Menu system

- Second, print the menu and use a selection statement:

```python
print("Main Menu")
print("1. Option 1")
print("2. Option 2")
choice = int(input("Choose an option: ")

if choice == 1:
   option1()
elif choice == 2:
   option2()
```

Menu system

- Third, add a validation loop:

```
print("Main Menu")
print("1. Option 1")
print("2. Option 2")
choice = int(input("Choose an option: ")

while choice != 1 and choice != 2:
    choice = int(input("Error, enter 1 or 2: "))

if choice == 1:
    option1()
elif choice == 2:
    option2()
```

Menu system

- Alternative validation loop (useful for lots of options):

```
print("Main Menu")
print("1. Option 1")
print("2. Option 2")
(print more options here)
choice = int(input("Choose an option: ")
while choice not in (1,2,3,4,5):
   choice = int(input("Error, enter 1 - 5: "))
if choice == 1:
   option1()
elif choice == 2:
   option2()
etc.
```

Worksheet 10c

- Complete **Worksheet 10c**

Plenary

- Explain the meaning of these terms:
  - Flag
  - Validation

Plenary Answers

- Explain the meaning of these terms:
  - Flag
    - A variable (usually a Boolean) is used to check whether something has happened
  - Validation
    - A method of checking data input in order to reject data that is obviously wrong

Copyright

© 2017 PG Online Limited

The contents of this unit are protected by copyright.

This unit and all the worksheets, PowerPoint presentations, teaching guides and other associated files distributed with it are supplied to you by PG Online Limited under licence and may be used and copied by you only in accordance with the terms of the licence. Except as expressly permitted by the licence, no part of the materials distributed with this unit may be used, reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic or otherwise, without the prior written permission of PG Online Limited.

Licence agreement

This is a legal agreement between you, the end user, and PG Online Limited. This unit and all the worksheets, PowerPoint presentations, teaching guides and other associated files distributed with it is licensed, not sold, to you by PG Online Limited for use under the terms of the licence.

The materials distributed with this unit may be freely copied and used by members of a single institution on a single site only. You are not permitted to share in any way any of the materials or part of the materials with any third party, including users on another site or individuals who are members of a separate institution. You acknowledge that the materials must remain with you, the licencing institution, and no part of the materials may be transferred to another institution. You also agree not to procure, authorise, encourage, facilitate or enable any third party to reproduce these materials in whole or in part without the prior permission of PG Online Limited.