Introduction to programming

Handling errors and testing

Exercise 1 – fix_errors.py

Your manager has provided you with the following piece of code from another member of the team:

```
import rand

random_numbers = []

size  = int(raw_input("How many numbers do you need?"))
limit = int(raw_input("Up to what limit?"))

for r in range(1, size)
    random_num = rand.randrange(limit)
    random_numbers.insert(len(random_numbers),random_num)
    print "Adding", random_num

random_number.sort()

print "Sorted numbers ->>"

index = 1
while True:
    print random_numbers[index]
    index += 1
    if index > len(random_numbers):
        break
```

The program is intended to ask a user how many random numbers they want generated within a certain range, sort them, and then display them back to the user. There are several problems with the code which you need to fix before the program will work as expected. Complete the changes and save the file as **fix_errors.py** in your Week 8 folder.

Extension: Ensure the program safely takes an integer number for the two questions when asked.

Exercise 2 – numbers_to_sort.py

Your line manager has asked you to write a program that opens a file, reads a list of numbers, sorts them and then prints them out. You can use the below code to open a file:

```
with open('file.txt', 'r') as f:
    # f will then hold the file object
```

You will need to extract the numbers from the file in to a list and then sort them. You will need to ensure that the program does not crash by catching exceptions if the file does not exist.

Save your file as **numbers_to_sort.py** in your Week 8 folder.

Exercise 3 – pick.py

The following code generates a list of random numbers which is itself a random length:

```
import random

list_of_numbers = []

for i in range(random.randrange(20)):
    list_of_numbers.append(random.randrange(100))

pick = int(raw_input("Pick a number you want to display:"))

print "You picked:", list_of_numbers[pick]
```

Add some exception handling to the program to ensure that the user doesn't pick the index of a number which is not in the list that has been generated. Test your program then save it as **pick.py** in your Week 8 folder.

Exercise 4 – Test plan

Pick one of your previous programs from last week for example the **file_reader.py** or **secured_login.py** program and write a test plan for it. You can approach this from a **black box** perspective.

Your test plan can be simple as per the details in the reading for this. Be sure to include at least a description of the test, the data, expected and actual results.

Once you have created your test plan, implement it against the program you selected making a note of any unexpected results.

Extension: If you have any unexpected results, create a second version of your program that addresses these issues and then re-check your test plan.