

GCSE Computer Science

Programming Code Bible

1	Error Checking	<pre>while a.isdigit() is False: print ("Invalid entry, number required") a=input(text)</pre>	Use a while loop to check whether there are any numbers in the variable called a. If it is false, that means that there is text in the variable print a suitable message and then ask for the input again. It will keep looping around these two lines of code until it is satisfied it is a digit and then will break out of this loop and carry on with the rest of the code.
2	Error Checking	<pre>while len(a)>6: print ("Invalid entry, number required") a=input(text)</pre>	Use a while loop to check whether the length of a is greater than 6. If it is it will print a suitable message and then ask for the input again. It will keep looping around these two lines of code until it is satisfied it is less than 6 and then will break out of this loop and carry on with the rest of the code.
3	Error Checking	<pre>while a != "FRANCE" and a!="AMERICA":</pre>	Will only run whatever is indented on the next line of code whilst this line of code is NOT true. != means not equal to
4	File Handling	<pre>f=open('gcse1.csv','wt',newline=")</pre>	This is needed before you start using a CSV file as it will open the CSV file. Replace gcse1 with the name of your file. Wt opens it in write mode which means if the file already exists it will clear the contents when it opens.
5	File Handling	<pre>import csv</pre>	Is needed at the start of the program when you are going to do anything with csv files anywhere in the program
6	File Handling	<pre>f=open('gcse1.csv','a',newline=")</pre>	This is needed when you are using a CSV file as it will open the CSV file. Replace gcse1 with the name of your file. A opens it in append mode which means if the file already exists it will add data to the current contents.
7	File Handling	<pre>writer = csv.writer(f)</pre>	You need this line of code once you have opened the CSV file but before you start writing data. The f needs to match the variable name you used when you opened the CSV file.
8	File Handling	<pre>writer.writerow((dicethrows))</pre>	Will write one row at a time the contents of dicethrows. In this

			instance dicethrows is the name of a list.
9	File Handling	f.close()	When you have finished reading or writing to a CSV file you must close it. The f needs to match the variable name you used when you opened the CSV file.
10	File Handling	csv_f = csv.reader(f)	You need this line of code once you have opened the CSV file but before you start reading data. The f needs to match the variable name you used when you opened the CSV file.
11	File Handling	for row in csv_f: newlist.append(row[0:])	To be used once a CSV file has been opened and read you can use this to populate a list with data from a CSV file. Newlist is the name of the empty list, append just means add to
12	File Handling	newlist = [] for row in csv_f: row[1]=int(row[1]) row[2]=int(row[2]) row[3]=int(row[3]) newlist.append(row[0:])	As each row of the csv is placed in the list it will convert to an integer
13	Functions	def inputnumber(text):	Needed at the start of the program to define the name of the function. This particular function is going to be called input number and the bit inside the brackets defines how many things you are expecting to be passed to the function when the main part of the code uses the function. In this particular example only one thing is going to be passed. To pass more than one thing you separate them with brackets.
14	Functions	return int(a)	This would appear at the end of a function. Functions always return back a value to the main part of the program. In this example the value of variable a is being returned. The returned value is also being type cast to an integer.
15	Functions	dice1=inputnumber("Enter die 1")	An alternative way to gather input is to use a function created. This will use the function called inputnumber and will pass the one thing inside the brackets to the function. The number of things inside the brackets must match the number of things inside the brackets where the

			function has been defined at the start of the code.
16	Input	<code>dice1=input ("Enter die 1")</code>	This will collect an input from the user. The text that appears on the screen is in the brackets and speech marks. The bit before the equals sign is the name of the variable the users input will be stored in.
17	Iteration	<code>for i in range (0,3):</code>	Will repeat whatever is indented under this line of code three times. i is just used as a letter to identify the counter. It can be any letter. Replace the number 3 for the number of times you wish to repeat. Don't forget the colon and make sure anything you wish to repeat is indented underneath and also when you want to stop repeating that you go back hard up against the left hand side.
18	Lists	<code>dicethrows=[dice1,dice2,dice3,score]</code>	This is a list that is called dicethrows. A list can hold more than one thing. This is currently holding the value of four variables. A list is defined by square brackets.
19	Lists	<code>newlist = []</code>	Creates an empty list ready to put data in.
20	Lists	<code>for i in wordlist: if i==word: print("Match found")</code>	Checks to see if an element in a list matches the value of a variable. The name of the list is word list and the name of the variable is word
21	Lists	<code>for i in range (0,10): coin=inputnumber("Enter a coin value") coinarray[i]=coin</code>	Will iteration through ten times. So i will be 0 the first time, 1 the second, 2 the third and so on. It will set the value of that position of the list to the value the user inputs and gets stored in the variable coin.
22	Lists	<code>PlayerName=["Helen","Adam","Lidia","Kwaku","Priya","Chan"] places=inputnumber("How many places to move?") for i in range(0,places): temp=PlayerName[5] PlayerName[5]=PlayerName[4] PlayerName[4]=PlayerName[3] PlayerName[3]=PlayerName[2] PlayerName[2]=PlayerName[1] PlayerName[1]=PlayerName[0] PlayerName[0]=temp</code>	Will shift the order of the list using a number that the user enters. It needs a temporary variable to hold the last item, it can then move them all up one, put the temporary value back in at the start and keep repeating this to all the moves are done.
23	Output	<code>print (score)</code>	This would output to the screen the current value of the variable score. To print out specific text simply put it in speech marks inside the brackets.

24	Output	scores = [[x[3], x[0],x[1],x[2]] for x in newlist]	This will take the 4 th , 1 st , 2 nd , and 3 rd items from whatever is in the list called newlist and put them in this list called scores. Remember the first element in Python is 0, and the one you want to eventually sort on needs to be put to the front. That is why the fourth element is at the front
25	Output	for entry in sorted(scores): print(entry)	Will sorted everything in the list called scores and print it out. It will sort on the first element in the list called scores.
26	Output	for entry in sorted(callog,reverse=True): print(entry)	The reverse = True parts puts the output in descending order
27	Selection	if dice1==dice2 and dice2==dice3: score=dice1+dice2+dice3	This will see if the values of variables dice 1 and dice 2 are the same as the values of variables dice 2 and dice 3. Don't forget the colon and don't forget to indent the next line of code and this tells it what to do if true. In this particular exam the value of the variable score will become the total of all three variables.
28	Selection	elif dice1==dice2: score = (dice1+dice2)-dice3	If the original if statement is not true then the program needs to know what else to check. An Elif is used to do this.
29	Selection	else: score=0	When all the conditions have been checked by using an if statement and then some elif statements if needed finally you need an else statement to capture any values or inputs that are not true of any of the previous conditions.
30	Type Casting	a=int(a)	This will convert the current value held in variable a to an integer and also store it back in the variable a.
31	Type Casting	a=a.upper()	This will convert the current value held in variable a to upper case and also store it back in the variable a.