**Average Module Grade Program**

**- Design**

The program will need to have variables for the mark that has just been entered (mark), how many marks have been entered so far (modules), the total of all marks (total) and the lowest mark (lowest). It will also need variables that track what grade boundaries each of the mark falls into (first, two1, second, third, fail) and three variables that I will use to end loops (valid, valid2, valid3).

I will also define a ‘rangeCheck’ function to check if the user has entered values been 0 – 100, which will be used for the initial entering in of marks, and the final input of the users desired average. It will have four variables: the value entered (val), the lowest number in the range (lo) and the highest number in the range(hi), and a string that displays a message if they enter an invalid number(msg). The function returns a boolean value, true if the number is valid, and false if it is invalid.

If the number is lower than the lowest

Print the message

Return false

If the number is higher than the highest

Print the message

Return false

Else

Return true

The number of modules that has been completed varies from person to person, so I need to use a loop that continues until it is told to stop. Inside the loop the user will input their mark, the program will check if it is within the range 0 – 100, with the rangeCheck function and if it is it will then add it to the total and work out which grade boundary it fits into.It will also see if it is lower than the current lowest mark, and if it is the new number replaces that mark.

While valid is false

Input new mark

Use range check function to see it is within 0 – 100

If it isn’t restart the loop

If it is continue:

If mark is lower than current lowest

Replace lowest with mark

Add one to the module counter

Unless it was a fail add the new mark to the total

Update the correct grade boundary

After this the user will be prompted if they want to enter another mark. If they type in ‘y’, the loop is set to false and repeats, if they enter ‘n’, it is set to true and continues. Using an embedded loop I can ensure that the user only enters ‘y’ or ‘n’ and if they enter anything else than they are asked again. If the user has already entered 16 modules, this is entirely disregarded and the program doesn’t go back into the first loop.

While valid2 is false

Set valid2 to True so the loop doesn’t start again

If less than 16 modules have been entered:

Ask the user if they want to continue

If user enters Y:

Set valid to false so the original loop ends

Else if the user enters N:

Set valid to true

Else if the user enters anything else

Tell the user they entered in something wrong

and restart the embedded loop

else if 16 modules have been entered,

set valid true so the original loop ends

Once the loop has finished display the how many of which grade boundaries the user got (e.g. two firsts, three 2:1s etc).

To disregard the lowest mark, take away the lowest from the total and take one away from the module counter.

To work out the averages I will need to use another loop, activated by valid3. This loop is disregarded if the user has entered 16 modules, and only their final average grade is displayed to them. If they’ve entered less than 16, then their current average is displayed and they are asked to enter the average they want when they finish their course (desAvg). If they enter an average lower than their current average they will also be prompted again. I will use the rangeCheck function to make sure the user enters a number between 0 and 100, if it fails then the loop starts again otherwise the program will then work out how many more marks they need to achieve their desired average, and then display it to the user.

if modules are less than 16

while valid3 is false

display current average

input desired average

Use range check function to see it is within 0 – 100

If it isn’t restart the loop

If it is continue:

modRem = 16 – modules

result = ((15\*desAvg)-total)/modules)

if desMark is less than total

print an error

restart the loop

else

print result and modRem

else print average (total/modules)

**- Testing**

1. (0,1,12,23,34,45,56,67,78,89,90,99,100,50,50,50)

(no desired average as 16 marks entered)

expected result: avg:51.6

actual result: avg:51.6

1. (0,1,12,23,34,45,56,67,78,89,90,99,100)

expected result: 52

actual result: 52

(70)

expected result: 117.75

actual result: 117.75

1. (0,12,34,-56,45,56,67,78)

expected result: program will not accept the negative number

actual result: program loops after negative number, entering any numbers after it loops affects total normally, negative number was disregarded.

1. (45,2,48,90,200,52,4)

expected result: program will not accept the number over 100

actual result: program goes into loop after 200 is entered, again the number does not affect the total.

1. (33,67,g,3,77,56,78)

expected result: after entering the letter g, the program will display an error then loop back to the input

actual result: as expected, no variables are affected by entering a string.

1. Entering (fhgjflkalsjdhf;lkasjdf;lkasjd) into mark input

Expected: the program will display an error and then restart the loop without affecting any variables

Actual: as expected

1. Entering (laksdhf;abfo;ans;lkasjdf;) into continue input

Expected: The program will display an error and then restart the loop without affecting any variables

Actual: as expected

1. Entering (asfasdfasfgwdgu340jlfhoeiruhf:) into desired average input

Expected: The program will display an error and then restart the loop without affecting any variables

Actual: as expected

1. Entering a desired average lower than currant average

Expected: The program will display an error and then restart the loop without affecting any variables

Actual: as expected