**2. Selection Problems (IF ELSE, ELIF)**

**Task 2:**

**Plan:**

Task 2 solution 1:

1. Declare list variable amount
2. Define and Validate float variable user, then append value to amount. If variable not a valid data type displays appropriate error message.
3. Define amount as its own sum.
4. Check if amount is bigger than 1000, then define amount = amount + 100
5. Else if check if amount is in range 1000 to 2000, then define amount = amount \* 2
6. Output (amount)

Task 2 solution 2:

1. Define variable valid = True
2. Validate through 3 different variables storing user input, and display appropriate error message when data type is wrong.
3. Define total as the sum of the 3 variables.
4. check if total is less than 1000, then define total = total + 100
5. else if check if total is in range from 1000 to 2000, then define total = total \*2
6. output (total)

**Pseudocode:**

Task 2 solution 1:

For I in range (1,4):

Try:

Print (“Enter amount for friend”, i)

User 🡸 input as float (“£ ”)

Amount.append (user)

Except:

Print (“not a valid amount”)

Amount 🡸 sum (amount)

If amount < 1000:

Amount 🡸 amount + 100

Else if amount in range (1000,2001):

Amount 🡸 amount \*2

Print ("Your total amount is £" + amount)

Task 2 solution 2:

Valid 🡸 True

While valid = True:

Try:

Friend1 🡸 input as float ("Enter amount for friend 1: £")

Friend2 🡸 input as float ("Enter amount for friend 2: £")

Friend3 🡸 input as float ("Enter amount for friend 3: £")

Valid 🡸 False

Except:

Print (error message)

Total 🡸 friend1 + friend2 + friend3

If total < 1000:

Total 🡸 total + 100

Else if total in range (1000,2001):

Total 🡸 total \*2

Print ("You have raised £" + total)

**variables table:**

Task 2 solution 1:

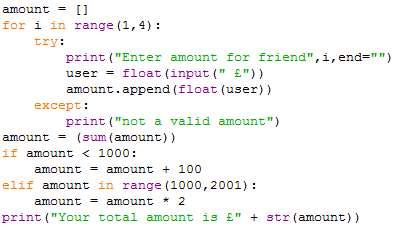
|  |  |  |
| --- | --- | --- |
| Variable name | Data type | Comment |
| Amount | List | User input is appended to it |
| User | Float | Store input from user. |

Task 2 solution 2:

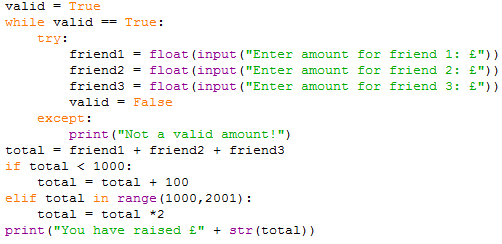
|  |  |  |
| --- | --- | --- |
| Variable name | Data type | Comment |
| Valid | Boolean | Stores Boolean value True |
| Friend1 | Float | Stores user input |
| Friend2 | Float | Stores user input |
| Friend3 | Float | Stores user input |
| Total | Float | Stores sum of friend1 + friend2 + friend3 |

**Screenshot evidence:**

Task 2 solution 1:

In the screenshot to the left as you can see, there is a for loop that loops through the variable user, and appends the value to a list variable amount. The for loop also validates the user variable and prints an appropriate error message when needed. Outside the for loop amount is defined with the sum of its self. Amount is then compared to 2 different conditions. The first condition checks if amount is less than 1000, the second checks if amount is in range from 1000 to 2000. Depending on what condition is chosen, the variable is then defined either with amount = amount + 100 or amount = amount \*2.

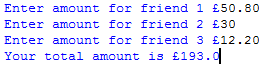
Task 2 solution 2:

in the screenshot to the left as you can see, variable valid is defined with a Boolean type of True. A while loop then checks the validation of 3 variables, and displays an error message if needed. After the while total is defined total = friend1 + friend2 + friend3. Total is then check against 2 conditions, the first being if total is less than 1000, then second being if total is in range 1000 to 2000. Total is then defined defending on the condition total = total + 100 for the first, or total = total \*2 for the second condition. Finally, total is output in a print statement.

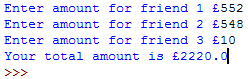
**Sample run (Output):**

Note: Both coded solutions produce the same output, so I have not put in a screenshot of both.

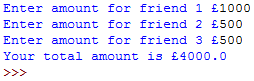
First condition) any amount raised less than £1000 has a £100 bonus:

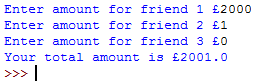
In the screenshot you can see that £100 has be successfully, added to the total value of the 3 friend’s money. This is show because 50.8 + 30 + 12.2 = 93, but the total value is 193. So the first condition has been met.

Second condition) the company will still double the amount raised between £1000 and £2000

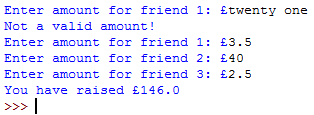
as you can see the second condition has been met, because 552 + 548 + 10 = 1110, but the total is 2220 so the value has been doubled.

If the amount is over £2000 the initial £2000 is doubled but any amount after that is not

in the screenshot to the left you can see that the total value of £2000 has been doubled, so the first part of the condition has been met. In the second screenshot bellow you can see that the total value of £2001 has not been doubled so the second part of the condition has been met.



Error message

as you can see that when I type a string for example “twenty one”, an error message appears saying “Not a valid amount!”. The user is then asked to type the value again. If the users types an integer or float the value is excepted.

**Flow charts Task 2 Solution 1:**

START

For I in range (4)

Loop

Print ("Enter amount for friend", i)

User 🡸 input as float (“£ ”)

Is user a float?

No

Print (“not a valid amount”)

Yes

Amount.append (user)

Amount 🡸 sum (amount)

No

Yes

END

Print ("Your total amount is £" + amount)

No

Amount 🡸 amount \* 2

Else if amount in range (1000,2001)

Yes

Amount 🡸 amount + 100

If amount < 1000:

**Flow charts Task 2 Solution 2:**

START

Print ("Not a valid amount!")

Friend1 🡸 input as float ("Enter amount for friend 1: £")

END

Print ("Your total amount is £" + amount)

Total 🡸 total \*2

Yes

Else if total in range (1000,2001)

No

Yes

Total 🡸 total + 100

If total < 1000:

Total 🡸 friend1 + friend2 + friend3

No

No

No

Yes

Yes

Yes

Is friend3 float?

Friend3 🡸 input as float ("Enter amount for friend 3: £")

Is friend2 float?

Is friend1 a float?

Friend2 🡸 input as float ("Enter amount for friend 2: £")