Week 3 – Control Structures

# Activity 1 – Breakdown Flowchart (Portfolio task)

Project name: breakdown\_flowchart

Ensure you complete this activity and add a screenshot of your flowchart ***breakdown\_flowchart.jpg*** to your week 3 folder (see Step 3 below the task)

Create a flowchart in a program of your choice (e.g. Microsoft word, Microsoft Visio, draw.io, Figma etc) for the breakdown activity your completed in week 1. See problem solving steps below to help you.

Decomposition:

Add explanatory comment

Declare variables with values

Perform calculations to work out profit and breakeven

Display the results

Prepare

breakdown.py

Act

Present

Algorithm:

1. Let item cost = 20.00
2. Let sale price = 30.00
3. Let fixed costs = 50000.00
4. profit = sale price – item cost
5. breakeven = fixed costs / (sale price – item cost)

Representative Algorithm name

- Breakeven Calculator

Pseudocode:

1. Start
2. Read item cost 🡪 input or as per the requirement spec
3. Read sale price 🡪 input or as per the requirement spec
4. Read fixed costs 🡪 input or as per the requirement spec
5. Compute and set profit 🡪 sale price – item cost
6. Compute and set breakeven 🡪 fixed costs / (sale price – item cost)
7. Write cost to produce each item
8. Write sale price for each item
9. Write fixed costs
10. Write profit per item
11. Write number of items to breakeven
12. End

Representative Algorithm name

- Breakeven Calculator

Step 1: Analyse the problem-solving steps above

* Read through the steps and review to the lecture slides

Step 2: Create a new document in Microsoft Word (or a similar programme of your choice):

* Using drawing shapes create a flowchart for the above problem-solving steps

Step 3: Take a screenshot of your flowchart

* Take a screenshot of your flowchart
* Paste the screenshot into Paint and save it in your week 3 folder as ***breakdown\_flowchart.jpg***.

## Activity 2 – University Grade Selector (Portfolio task)

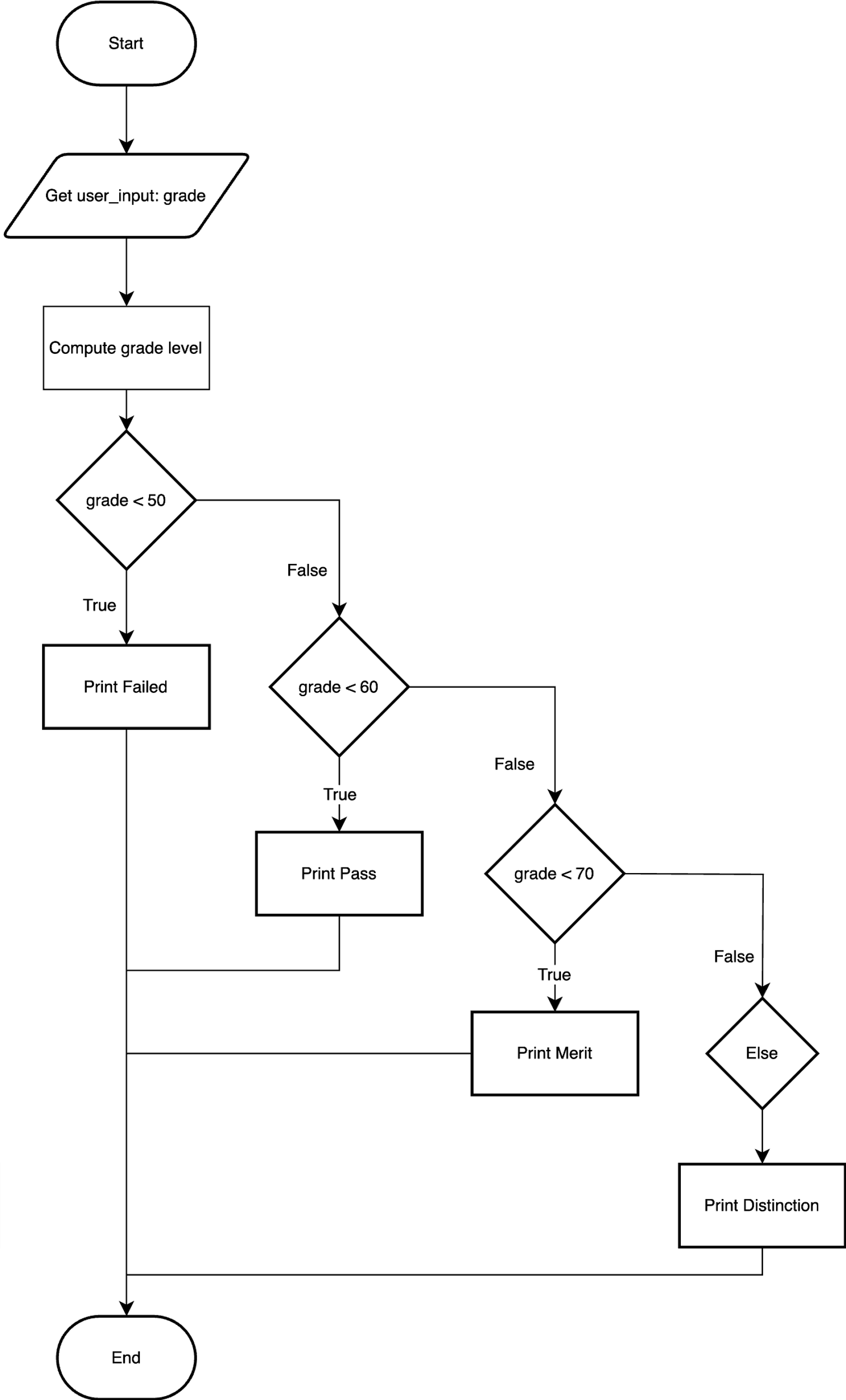
Project name: selection\_grades

Ensure you complete this task and add the source code ***selection\_grades.py*** and screenshot of the output to your week 3 folder (see Step 5 below the task)

Step 1: Create a new Python script file

* Create a new file called ***selection\_ grades.py*** in your week 3 folder

Step 2: Analyse the flowchart below:



Step 3: Write the Python code using the flowchart provided in Step 2

* Ensure to use user input and selection (relational and Boolean operators, and if and else, where needed)
* Output the result with a clear label

Step 4: Run your program

* Does your program run? If not, what errors do you have? Look at previous examples to see if you’ve used the correct code

Step 5: Take a screenshot of the output

* Take a screenshot of the application window showing the program output
* Paste the screenshot into Paint and save it in your week 3 folder as ***selection\_grades.jpg***.

## Activity 3 – Traffic Lights with Turtle (Portfolio task)

Project name: traffic\_lights

Ensure you complete this task and add the source code ***traffic\_lights.py*** and screenshot of the output to your week 3 folder (see Step 4 below the task)

Step 1: Create a new Python script file

* Create a new file called ***traffic\_lights.py*** in your week 3 folder
* Copy and paste the code below into your new file

import turtle

# Set up the screen

wn = turtle.Screen()

wn.title("Traffic Light Simulation")

wn.bgcolor("darkgrey")

# Create a turtle for drawing the traffic lights

light = turtle.Turtle()

light.shape("circle")

*def* update\_light\_color(*color*):

   # Complete the code: Set the light color

# Event handlers for keypress to change the traffic light colors

*def* go\_green():

    update\_light\_color("green")

*def* slow\_down():

    update\_light\_color("yellow")

*def* stop():

    update\_light\_color("red")

# Complete the code: Bind the keypress to the event handlers

# Listen to the keyboard events

wn.listen()

# Keep the window open until it's closed by the user

wn.mainloop()

Step 2: Complete the missing code – Look at the comments for the following lines and add the code that’s missing

* Complete the code: Set the light color
* Complete the code: Bind the keypress to the event handlers

Step 3: Run your program

* Does your program run? If not, what errors do you have? Look at previous examples to see if you’ve used the correct code

Step 4: Take a screenshot of the output

* Take a screenshot of the application window showing the program output
* Paste the screenshot into Paint and save it in your week 3 folder as ***traffic\_lights.jpg***.

## Activity 4 – Weight Loss Calculator Test Plan

Project name: weight\_loss

Ensure you complete this task and add the source code ***weight\_loss.py*** and screenshot of the output to your week 3 folder (see Step 4 below the task)

Step 1: Create a new Python script file

* Create a new file called ***weight\_loss.py*** in your week 3 folder
* Copy and paste the code below into your new file

# Calculate weight lost from calories burnt

CYCLE\_CALS\_PER\_HOUR = 100

RUNNING\_CALS\_PER\_HOUR = 475

SWIMMING\_CALS\_PER\_HOUR = 275

cycle\_hours = float(input("Please enter hours spent cycling: "))

running\_hours = float(input("Please enter hours spent running: "))

swimming\_hours = input("Please enter hours spent swimming: ")

total\_cycle\_cals = cycle\_hours + CYCLE\_CALS\_PER\_HOUR

total\_running\_cals = running\_hours \* RUNNING\_CALS\_PER\_HOUR

total\_swimming\_cals = swimming\_hours \* SWIMMING\_CALS\_PER\_HOUR

total\_cals\_burnt = total\_cycle\_cals + total\_running\_cals + total\_swimming\_cals

total\_pounds\_lost = total\_cals\_burnt / 3500

print("You have burnt {*:.2f*} calories and lost {*:.2f*} pounds"

      .format(total\_cals\_burnt, total\_pounds\_lost))

Step 3: Run your program 3 times

* Enter the values when prompted into the terminal window to complete the test plan below:

|  |  |  |  |
| --- | --- | --- | --- |
| **Input** | **Reason** | **Expected** | **Actual** |
| Cycling = 0  Running = 0  Swimming = 0 | all input values of 0 should result in output values of 0 | (0 \* 200) + (0 \* 475) + (0 \* 275) = 0 cals  0 / 3500 = 0 pounds |  |
| Cycling = 10  Running = 2  Swimming = 2 | calories burnt is 3500 which should result in 1 pound loss | (10 \* 200) + (2 \* 475) + (2 \* 275) = 3500 cals  3500 / 3500 = 1 pound |  |
| Cycling = 1.5  Running = 2.25  Swimming = 3.75 | all input values different | (1.5 \* 200) + (2.25 \* 475) + (3.75 \* 275) = 2637.5 cals  2637.5 / 3500 = 0.75 pounds (to 2 decimal places) |  |

Step 4: Identify the errors in the code

* Find the errors in the code
* Fix the errors

Step 5: Run your program 3 times again until you have the correct answers

* Enter the values when prompted into the terminal window to complete the test plan below:

|  |  |  |  |
| --- | --- | --- | --- |
| **Input** | **Reason** | **Expected** | **Actual** |
| Cycling = 0  Running = 0  Swimming = 0 | all input values of 0 should result in output values of 0 | (0 \* 200) + (0 \* 475) + (0 \* 275) = 0 cals  0 / 3500 = 0 pounds |  |
| Cycling = 10  Running = 2  Swimming = 2 | calories burnt is 3500 which should result in 1 pound loss | (10 \* 200) + (2 \* 475) + (2 \* 275) = 3500 cals  3500 / 3500 = 1 pound |  |
| Cycling = 1.5  Running = 2.25  Swimming = 3.75 | all input values different | (1.5 \* 200) + (2.25 \* 475) + (3.75 \* 275) = 2637.5 cals  2400.00 / 3500 = 0.69 pounds (to 2 decimal places) |  |

Step 6: Take a screenshot of the output

* Take a screenshot of the terminal window showing the program output
* Paste the screenshot into Paint and save it in your week 3 folder as ***weight\_loss.jpg***.