Module 3: Control Structures

Premanand S

Assistant Professor,
School of Electronics and Engineering,
Vellore Institute of Technology, Chennai

premanand.s@vit.ac.in

August 20, 2024

Topics to be covered in Module 3,

- Decision making and Branching
- if, if – else, nested if, multi-way if-elif statements
- Looping - While loop, For loop, else clauses in the loop,
- Nested loop
- Break, Continue, and Pass Statements

Example 1: Pizza delivery

Example (Python Snippet)

Write a Python program for a pizza delivery service that calculates the total bill based on the size of the pizza, whether the customer wants pepperoni, and if they want extra cheese. The program should prompt the user to enter their choices and then calculate the final bill amount. The costs are as follows: Small pizza costs 150, Medium pizza costs 175, and Large pizza costs 200. Adding pepperoni costs 20 for a small pizza and 30 for a medium or large pizza. Adding extra cheese costs 25 regardless of the pizza size. Display the final bill amount to the user.

Example 1: Pizza delivery

Example 1: Pizza delivery

Example (Python Snippet)

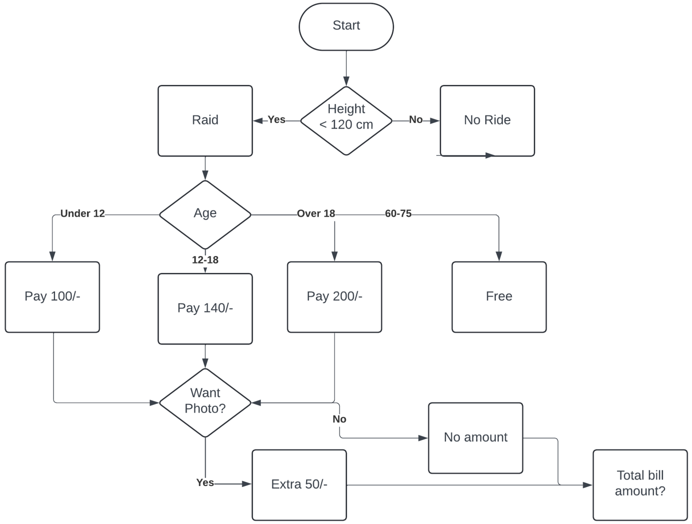
```
print('Thank you for choosing python pizza delivery!')
size = input('What pizza you need?(s/m/l)')
add_pepperoni = input('You need pepperoni?(y/n)')
add_cheese = input('you need extra cheese?(y/n)')
bill = 0
if size =='s':
    bill += 150
elif size == 'm':
    bill += 175
else:
    bill += 200
```

Example 1: Pizza delivery

Example (Python Snippet)

```
if add_pepperoni ==’y’:
    if size==’s’:
        bill +=20
    else:
        bill +=30
if add_cheese ==’y’:
    bill +=25
print(f’Your final bill is:{bill}’)
```

Example 2: Roller Coaster - Senior Citizen



Example 2: Roller Coaster - Senior Citizen

Example (Python Snippet)

```
print('Welcome to the roller coaster!')
height = int(input('What\'s your height in cm? '))
bill = 0

if height >= 120:
    print('You can ride!')
    age = int(input('What\'s your age? '))

    if age < 12:
        bill = 100
        print('Please pay Rs:100/-')
    elif age <= 18:
        bill = 140
        print('Please pay Rs:140/-')
```

Example 2: Roller Coaster - Senior Citizen

Example (Python Snippet)

```
    elif 60 <= age <= 75:
        bill = 0
        print('Ticket is free for your age group (60-75)!')
    else:
        bill = 200
        print('Please pay Rs:200/-')

    photo = input('Do you need a photo? Yes/No ').lower()

    if photo == 'yes':
        bill += 50

    print('Your final bill amount:', bill)
else:
    print('You cannot ride!')
```

Try and Except

- try and except blocks are used for handling exceptions, which are errors that occur during the execution of a program.
- The try block lets you test a block of code for errors, while the except block lets you handle the error.

try, except, else, and finally

Example (Python Snippet)

```
try:
    # Code that might raise an exception
    pass
except ExceptionType as e:
    # Code to handle the exception
    pass
else:
    # Code to execute if no exception occurs
    pass
finally:
    # Code to execute regardless of whether an exception
    # occurs or not
    pass
```

Importance of Handling a Specific Exception

Example (Python Snippet)

```
result = 10 / 0
print(result)
```

Example 1: Handling a Specific Exception

Example (Python Snippet)

```
try:
    result = 10 / 0
except ZeroDivisionError:
    print("Cannot divide by zero!")
```

Example 2: Handling Multiple Exceptions

Example (Python Snippet)

```
try:
    num = int(input("Enter a number: "))
    result = 10 / num
except ZeroDivisionError:
    print("Cannot divide by zero!")
except ValueError:
    print("Invalid input! Please enter a number.")
```

Example 3: Handling Multiple Exceptions

Example (Python Snippet)

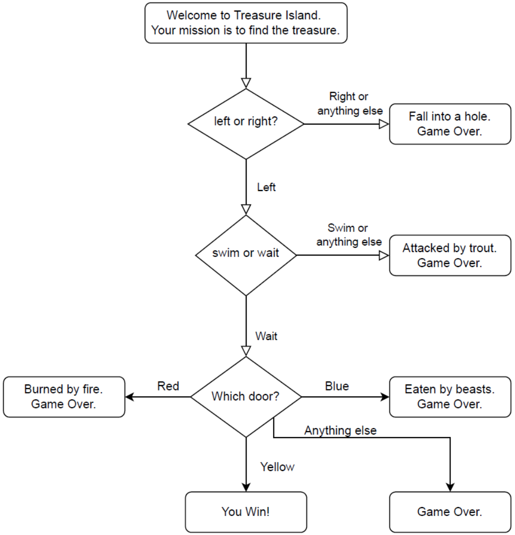
```
try:
    # Code that might raise different types of exceptions
    result = 10 / int(input("Enter a number: "))
except (ValueError, ZeroDivisionError) as e:
    # Handle multiple types of exceptions
    print(f"Error: {e}")
```

Example 4: Using else and finally

Example (Python Snippet)

```python
try:
    num = int(input("Enter a number: "))
    result = 10 / num
except ZeroDivisionError:
    print("Cannot divide by zero!")
except ValueError:
    print("Invalid input! Please enter a number.")
else:
    print(f"Result is {result}")
finally:
    print("Execution complete.")
```

PAT


Welcome to Treasure Island.
Your mission is to find the treasure.
left or right?
Right or anything else
Fall into a hole. Game Over.
Left
swim or wait
Swim or anything else
Attacked by trout. Game Over.
Wait
Which door?
Red
Burned by fire. Game Over.
Blue
Eaten by beasts. Game Over.
Anything else
Yellow
You Win!
Game Over.

PAT

Example (Python Snippet)

```
print('Welcome to Treasure Island game!')
print('Your mission is to find the treasure.')

choice1 = input('You\'re at a cross road. Where do
you want to go? Type "left" or "right" \n').lower()
if choice1 == "left":
  choice2 = input('You\'ve come to a lake. There
  is an island in the middle of the lake. Type "wait"
  to wait for a boat. Type "swim" to
  swim across. \n').lower()
  if choice2 == "wait":
    choice3 = input("You arrive at the island
    unharmed. There is a house with 3 doors. One red,
    one yellow and one blue. Which colour do you choose? \n")
```

PAT

Example (Python Snippet)

```
    if choice3 == "red":
      print("It’s a room full of fire. Game Over.")
    elif choice3 == "yellow":
      print("You found the treasure! You Win!")
    elif choice3 == "blue":
      print("You enter a room of beasts. Game Over.")
    else:
      print("You chose a door that doesn’t exist. Game Over.")

  else:
    print("You get attacked by an angry trout. Game Over.")
else:
  print("You fell into a hole. Game Over.")
```

Module

- A module is a file that contains Python code, which can include definitions of functions, classes, variables, and runnable code.
- Modules are used to organize code into manageable sections, allowing for better structure and reusability.
- By grouping related code into a module, developers can simplify complex programs and avoid redundancy.

Module - Types

- Built-in modules
- User defined modules

Creating a module

Example (Python Snippet)

```python
# the following code can be saved in a file named calc.py
def add(x, y):
    return x + y

def subtract(x, y):
    return x - y
```

Importing module

Example (Python Snippet)

```
import calc

result = calc.add(10, 5)
print(result)
```

Importing module

Example (Python Snippet)

```
from calc import add

result = add(10, 5)
print(result)
```

Importing module

Example (Python Snippet)

```
import calc as c

result = c.subtract(10, 5)
print(result)
```

Random Module

- The random module in Python is a built-in module that provides various functions to generate random numbers, choose random elements from a sequence, and perform random permutations, among other tasks.
- It's commonly used in simulations, games, security, and anywhere randomness is needed.

Random Module

Example (Python Snippet)

```
import random
print(dir(random))
```

Random Module - Documentation

Example (Python Snippet)

```
import random
help(random)
```

Random Module - random

Example (Python Snippet)

```
#Returns a random floating-point number between 0.0
(inclusive) and 1.0 (exclusive)

import random
random_float = random.random()
random_float
```

Random Module - random

Example (Python Snippet)

```
import random
random_float1 = [random.random() for _ in range(5)]
print(random_float1)
```

Random Module - random

Example (Python Snippet)

```
import random
random_float2 = random.random() * 5
print(random_float2)
```

Random Module - random

Example (Python Snippet)

```
import random
a = 10
b = 20
random_float3 = a + (b-a) * random.random()
print(random_float3)
```

Random Module - uniform

Example (Python Snippet)

```
#Returns a random floating-point number between a and
b (inclusive of a and b).

import random
random_float4 = random.uniform(10.3, 20.3)
print(random_float4)
```

Random Module - uniform

Example (Python Snippet)

```
#Returns a random floating-point number between a and
b (inclusive of a and b).

import random
random_float_5 = [random.uniform(0,50) for _ in range(10)]
print(random_float_5)
```

Random Module - randint

Example (Python Snippet)

```
#Returns a random integer between a and b (inclusive).
import random
die_roll = random.randint(1,6)
print(die_roll)
```

Random Module - randrange

Example (Python Snippet)

```
#Returns a randomly selected element from the range created
by start, stop, and step.

import random
random_num1 = random.randrange(10)
print(random_num1)
```

Random Module - randrange

Example (Python Snippet)

```
import random
random_num2 = random.randrange(10, 20)
print(random_num2)
```

Random Module - randrange

Example (Python Snippet)

```
import random
random_num3 = random.randrange(0,20,2)
print(random_num3)

import random
random_num4 = random.randrange(0,20,3)
print(random_num4)
```

Random Module - choice

Example (Python Snippet)

```
#Returns a random element from a non-empty sequence
(like a list or tuple).

import random
schools = ['SENSE', 'SELECT', 'SCOPE']
print(random.choice(schools))
```

Random Module - choices

Example (Python Snippet)

```
#Returns a list with k randomly selected elements from
the population. The weights or cum_weights can be used to
influence the probability of each element being chosen.

import random
students = ['keerthana', 'thusyanthan', 'vidya',
        'premanand', 'prem', 'anand']
weights = [1, 10, 5, 20, 25, 30]
task1 = random.choices(students, weights=weights, k=5)
print(task1)
```

Random Module - shuffle

Example (Python Snippet)

```
#Shuffles the sequence x in place. This modifies
the original list.



import random
cards = ['1','2','3','4','5','6','7','8','9','10',
                            'J','Q','K','Jo']
random.shuffle(cards)
print(cards)
```

Random Module - sample

Example (Python Snippet)

```
#Returns a list of k unique elements chosen
from the population.

import random
foodie = ['briyani','puttu','fish','mutton','chicken',
'squid','fried rice', 'rabbit','kadai', 'octobus','duck']
meals = random.sample(foodie, k=5)
print(meals)
```

Random Module - seed

Example (Python Snippet)

```
#Initializes the random number generator.
If a is provided, it ensures reproducibility of the
sequence of random numbers.



import random
random.seed(42)
print('with seed!')
for _ in range(10):
    print(random.random())
```

Random Module - without seed

Example (Python Snippet)

```
import random
print('without seed!')
for _ in range(10):
    print(random.random())
```

Problem 1: Random module - Love Score

Example (Python Snippet)

```
How you can calculate the love score with the help of
random module?
```

Problem 1: Random module - Love Score

Example (Python Snippet)

```
import random
love_score = random.randint(1,100)
print(f'Your love score is {love_score}')
```

Problem 2: Random module - Tossing a coin

Example (Python Snippet)

`How can you toss a coin like heads or tails using random module?`

Problem 2: Random module - Tossing a coin

Example (Python Snippet)

```
import random
random_side = random.randint(0,1)
if random_side == 1:
    print('Heads')
else:
    print('Tails')
```