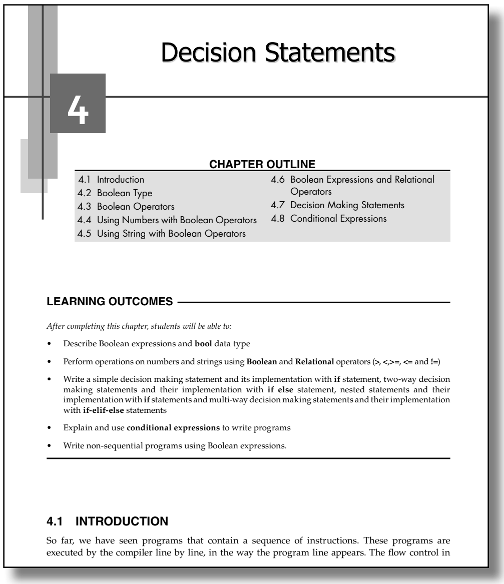
Visual Walkthrough

All chapters within the book have been structured into the following important pedagogical components:

- **Learning Outcomes** give a clear idea to the students and programmers on what they will learn in each chapter. After completion of chapter, they will able to comprehend and apply all the objectives of the chapter.

- **Introduction** explains the basics of each topic and familiarizes the reader to the concept being dealt with.



Decision Statements

4

CHAPTER OUTLINE

4.1 Introduction
4.2 Boolean Type
4.3 Boolean Operators
4.4 Using Numbers with Boolean Operators
4.5 Using String with Boolean Operators
4.6 Boolean Expressions and Relational Operators
4.7 Decision Making Statements
4.8 Conditional Expressions

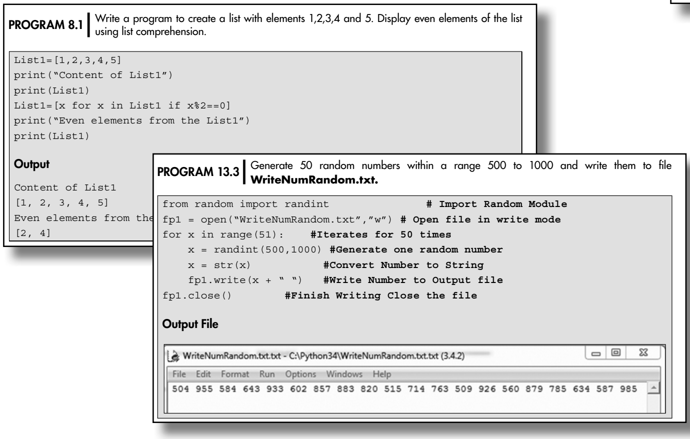
LEARNING OUTCOMES

After completing this chapter, students will be able to:

- Describe Boolean expressions and **bool** data type
- Perform operations on numbers and strings using **Boolean** and **Relational** operators (>, <,>=, <= and !=)
- Write a simple decision making statement and its implementation with **if** statement, two-way decision making statements and their implementation with **if else** statement, nested statements and their implementation with **if** statements and multi-way decision making statements and their implementation with **if-elif-else** statements
- Explain and use **conditional expressions** to write programs
- Write non-sequential programs using Boolean expressions.

4.1 INTRODUCTION

So far, we have seen programs that contain a sequence of instructions. These programs are executed by the compiler line by line, in the way the program line appears. The flow control in



PROGRAM 8.1 Write a program to create a list with elements 1,2,3,4 and 5. Display even elements of the list using list comprehension.

```
List1=[1,2,3,4,5]
print("Content of List1")
print(List1)
List1=[x for x in List1 if x%2==0]
print("Even elements from the List1")
print(List1)
```

Output

```
Content of List1
[1, 2, 3, 4, 5]
Even elements from the
[2, 4]
```

PROGRAM 13.3 Generate 50 random numbers within a range 500 to 1000 and write them to file **WriteNumRandom.txt.**

```
from random import randint               # Import Random Module
fp1 = open("WriteNumRandom.txt","w") # Open file in write mode
for x in range(51):    #Iterates for 50 times
    x = randint(500,1000) #Generate one random number
    x = str(x)           #Convert Number to String
    fp1.write(x + " ")   #Write Number to Output file
fp1.close()        #Finish Writing Close the file
```

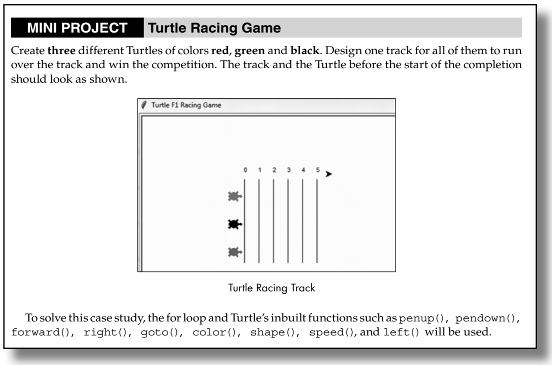
Output File

WriteNumRandom.txt.txt - C:\Python34\WriteNumRandom.txt.txt (3.4.2)
File Edit Format Run Options Windows Help

```
504 955 584 643 933 602 857 883 820 515 714 763 509 926 560 879 785 634 587 985
```

- **Programs** are the highlighting feature of the chapters. Ample programs have been provided against each sub topic to effectively strengthen the learnt concepts.

- **Mini Project** consists of a problem statement that will compel the readers to think and make use of various concepts learnt to solve real-life problems through programming.



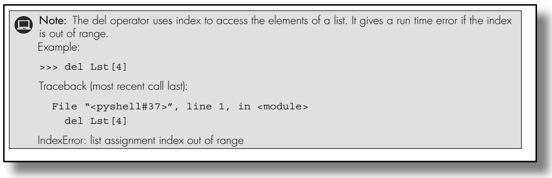
MINI PROJECT **Turtle Racing Game**

Create **three** different Turtles of colors **red**, **green** and **black**. Design one track for all of them to run over the track and win the competition. The track and the Turtle before the start of the completion should look as shown.

Turtle Racing Track

To solve this case study, the for loop and Turtle's inbuilt functions such as `penup()`, `pendown()`, `forward()`, `right()`, `goto()`, `color()`, `shape()`, `speed()`, and `left()` will be used.

- **Notes** have been inserted in each chapter to provide valuable insights based on programming concepts. Notes shall also act as precautionary statements for readers to solve programming problems effectively.



Note: The del operator uses index to access the elements of a list. It gives a run time error if the index is out of range.

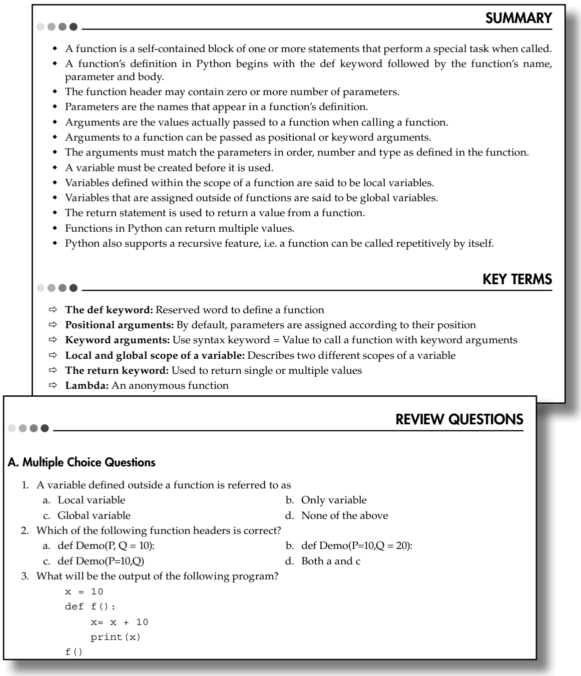
Example:

```
>>> del Lst[4]

Traceback (most recent call last):

  File "<pyshell#37>", line 1, in <module>
    del Lst[4]

IndexError: list assignment index out of range
```

- A concise **Summary** has been listed at chapter-end to reiterate vital points and describes in short, the complex concepts covered within the chapter.

- **Key Terms** enlists important keywords and concepts covered within the chapter.

- Extensive **Review Questions** presented at the end of each chapter comprise Multiple Choice Questions, True False statements, Exercise Questions and Programming Assignments. This would help in analyzing the learnt information.



SUMMARY

- A function is a self-contained block of one or more statements that perform a special task when called.
- A function's definition in Python begins with the def keyword followed by the function's name, parameter and body.
- The function header may contain zero or more number of parameters.
- Parameters are the names that appear in a function's definition.
- Arguments are the values actually passed to a function when calling a function.
- Arguments to a function can be passed as positional or keyword arguments.
- The arguments must match the parameters in order, number and type as defined in the function.
- A variable must be created before it is used.
- Variables defined within the scope of a function are said to be local variables.
- Variables that are assigned outside of functions are said to be global variables.
- The return statement is used to return a value from a function.
- Functions in Python can return multiple values.
- Python also supports a recursive feature, i.e. a function can be called repetitively by itself.

KEY TERMS

⇨ **The def keyword:** Reserved word to define a function
⇨ **Positional arguments:** By default, parameters are assigned according to their position
⇨ **Keyword arguments:** Use syntax keyword = Value to call a function with keyword arguments
⇨ **Local and global scope of a variable:** Describes two different scopes of a variable
⇨ **The return keyword:** Used to return single or multiple values
⇨ **Lambda:** An anonymous function

REVIEW QUESTIONS

A. Multiple Choice Questions

1. A variable defined outside a function is referred to as
   a. Local variable b. Only variable
   c. Global variable d. None of the above
2. Which of the following function headers is correct?
   a. def Demo(P, Q = 10): b. def Demo(P=10,Q = 20):
   c. def Demo(P=10,Q) d. Both a and c
3. What will be the output of the following program?

```
x = 10
def f():
    x= x + 10
    print(x)
f()
```