# TITLE: CONDITIONAL STATEMENTS AND BRANCHING

**OBJECTIVE**:

* To understand Conditional statements in python.

**THEORY**:

The statements that are used to control the flow of program based on certain conditions are called conditional statements. These statements allow the program to make decisions and execute different blocks of code depending on whether a specified condition evaluates to true or false. In Python, the most commonly used conditional statements are the **if**, **elif** (short for "else if"), and **else** statements. Conditional statements are specific syntax structures (like if, elif, else) used to implement branching in programming. Branching is the broader concept that refers to the program's ability to take different paths or branches based on conditions. Therefore, conditional statements are a mechanism for implementing branching in code.

1. **Simple if statement:**

It is the simplest form of conditional statement in Python is the if statement. It is used to execute a block of code only if a specified condition is true.

Syntax:

|  |
| --- |
| if condition:  # Code to execute if the condition is true |

1. **If-else statement:**

The if-else statement extends the if statement by providing an alternative block of code to execute when the condition is false.

Syntax:

|  |
| --- |
| if condition:  # Code to execute if the condition is true  else:  # Code to execute if the condition is false |

1. **If-elif-else chain:**

The if-elif-else chain is used when there are multiple conditions to check.

It evaluates each condition sequentially and executes the block of code associated with the first true condition.

Syntax:

|  |
| --- |
| if condition\_1:  # Code to execute if condition\_1 is true  elif condition\_2:  # Code to execute if condition\_2 is true  else:  # Code to execute if none of the conditions are true |

1. **Multiple elif blocks:**

We can have multiple elif blocks to handle various conditions. Each elif is checked only if the conditions preceding it are false.

Syntax:

|  |
| --- |
| if condition\_1:  # Code to execute if condition\_1 is true  elif condition\_2:  # Code to execute if condition\_2 is true  elif condition\_3:  # Code to execute if condition\_3 is true  #………………….. more elif conditions  else:  # Code to execute if none of the conditions are true |

1. **Testing multiple conditions:**

Sometimes, we need to test multiple conditions in a single if statement. In python, this can be done by using logical operators (and, or, not).

Syntax:

|  |
| --- |
| if condition\_1 and condition\_2:  # Code to execute if both conditions are true |

1. **Switch case:**

|  |
| --- |
| def switch\_case(expression):  if expression == value1:  # code to be executed if expression == value1  elif expression == value2:  # code to be executed if expression == value2  # additional elif cases...  else:  # code to be executed if none of the cases match |

The switch statement is a conditional statement used to perform different actions based on the value of an expression. It provides a way to streamline multiple if-else statements, especially when there are many possible values to check. Earlier, switch case was performed using if-elif- chain as:

Onwards python 3.10, switch\_case is achieved by using match and case keywords.

Syntax:

|  |
| --- |
| match term:  case pattern-1:  action-1  case pattern-2:  action-2  case pattern-3:  action-3  case \_:  action-default |

1. #Qn 1

2. #print all the even numbers from 1 to 10.

3. for i in range(1,10):

4.     if i%2==0:

5.         print (i)

6.

7. #Qn: 2

8. '''

9. #2 Add logic to print two lines.

10. The first line should contain the result of integer division //.

11. The second line should contain the result of float division.

12. '''

13. a=int(input("First Number: "))

14. b=int(input("Second Number: "))

17. print("The integer division is:",(a//b))

18. print("The float division i: ",(a/b))

19.

20. #Qn 3 Evaluate:

21. #a. 4\*(6+5)

22. print(4\*(6+5))

23. #b. 4\*6+5

24. print(4\*6+5)

25. #c. 4+6\*5

26. print(4+6\*5)

27. #d. (5>4) and (5==5)

28. print((5>4)and (5==5))

29. #e. not (5>4)

30. print(not (5>4))

31. #f. (5>4) or (3==5)

32. print((5>4) or (3==5))

33. #g. not ((5>4)or(3==5))

34. print(not ((5>4)or(3==5)))

35. #h. (True and True)and(True ==False)

36. print((True and True)and(True ==False))

37. #i. (not False) or (not True)

38. print((not False) or (not True))

39.

40. #Qn 4. What is the type of the result of the expression 3\*1.5+4

41. print(type(3\*1.5+4))

42.

43. #Qn 5. Find the numbers square root and square

44. number=4

45. print("The square root is",4\*\*0.5)

46. print("The square is:",4\*\*2)

47.

48. #Qn 6. Take an input from user then reverse the string using slicing:

49. user\_string=str(input("Enter String:"))

50. print("Reversed String is:",user\_string[::-1])

51.

52. #Qn 7. Write code to input from user and store it in variavle spam and then print Hello if 1 is stored in spam, print Hi, if 2 is stored in spam ad print Greetings! if anything else is stored in spam.

53. user\_input=int(input("Prompt Number:"))

54. if user\_input==1:

55.     print("Hello")

56. elif user\_input==2:

57.     print("Hi")

58. else:

59.     print("Greetings !")

60.

61. #Qn 8. Write a python script that takes two numbers as input and print their sum, difference, product and quotient using match case.

62. num\_1=int(input("First Number:"))

63. num\_2=int(input("Second Number:"))

64. operator=input("Operation: ")

65. match operator:

66.     case '+':

67.         print("The sum is: ", num\_1 + num\_2)

68.     case '-':

69.         print("The difference is:", num\_1-num\_2 )

70.     case '\*':

71.         print("The product is:", num\_1\*num\_2 )

72.     case '/':

73.         print("The float division is:", num\_1/num\_2 )

74.     case '//':

75.         print("The quotient is:", num\_1//num\_2 )

76.     case \_:

77.         print("Invalid operation !!")

78.

79. #Qn 9. Write a script that asks a user for the name and age, then prints a message that tells them the year in which they will turn 100 years old.

80. stringName=str(input("Enter your Name:"))

81. stringAge=int(input("Enter your Age:"))

82. print("Hello",stringName,"!! You will be 100 years old in",2023+(100-stringAge))

83.

84. # Qn 10. Create a python script that converts temperature from Fahrenheit to Celcius and viceversa.

85. temperature=int(input("Input Temperature:"))

86. stringUnit=str(input("Temp. Degree ?? (C/F):"))

87. if(stringUnit=='C'or stringUnit=='c'):

88.     print("In Farenheit, it Equals:",(9/5\*temperature)+32,"degree.")

89. elif(stringUnit=='F'or stringUnit=='f'):

90.     print("In Celcius. it is Equals:",(temperature-32)\*5/9,"degree.")

91. else:

92.     print("Invalid Conversion !!")

93.

94. #Qn 11. Create a program that asks for a age and prints "Child" if the age is less than 12, "Teenager" if age is between 13 and 19 and "Adult" for ages 20 and above.

95.

96. string\_age=int(input("Enter your age:"))

97. if string\_age<=12:

98.     print("You are Child !")

99. elif string\_age>=13 and string\_age<=19:

100.     print("You are a teenager !")

101. elif string\_age>=20:

102.     print("You are an adult !")

103. else:

104.     print("Something went wrong !")

105.

106. #Qn 12. Write a python script that takes a letter grade (A,B,C,D,F) as input and prints the corresponding grade point average (GPA). Eg. A=4.0, B=3.0, C=2.0, D=1.0, F=0.0. Include an else statement to handle invalid inputs.

107. stringGrade=str(input("Enter the grade: "))

108. if stringGrade=='A' or stringGrade=='a':

109.     print("Equivalent Grade Point (GP)= 4.0")

110. elif stringGrade=='B' or stringGrade=='b':

111.     print("Equivalent Grade Point (GP)= 3.0")

112. elif stringGrade=='C' or stringGrade=='c':

113.     print("Equivalent Grade Point (GP)= 2.0")

114. elif stringGrade=='D' or stringGrade=='d':

115.     print("Equivalent Grade Point (GP)= 1.0")

116. elif stringGrade=='F' or stringGrade=='f':

117.     print("Equivalent Grade Point (GP)= 0.0")

118. else:

119.     print("Invalid Grade Entered !!")

120.

121. #Qn 13. Wtrite a python program that takes a number and print whether it is "Even", "Odd", "Zero" or "Invalid" for non-integer inputs. This program should first check if the input is valid integer and then only check for other conditions.

122. numberInput=input("Enter the number")

123. if numberInput.isdigit:

124.     calcValue=int(numberInput)

125.     if(calcValue%2==0):

126.         print("It is even number")

127.     elif(calcValue%2!=0):

128.         print("It is odd number")

129.     else:

130.         print("It is zero !")

131. else:

132.     print("It is not a valid integer !!")

133.

134. Qn 14. . An extra day is added to the calendar almost every four years as February 29, and the thay is called a leap day. It corrects the calendar for the fact that our planet takes approximately 365.25 days to orbit the sun. A leap year contains a leap day. In the Gregorian calendar, three conditions are used to identify leap years.

135. • The year can be evenly divided by 4, is a leap year, unless

136. • The year can be evenly divided by 100, it is not a leap year, tunless:

137. • The year is also evenly divisible by 400. Then it is a leap year

138. This means that in the Gregorian calendar, the years 2000 and 2400 are leap years, while 1800, 1900, 2100, 2200, 2300 and 2500 are not leap years.

139. year=int(input("Enter the year: "))

140. print("Leap Year Check ",end=":")

141. if ((year%4==0)):

142.     if(year%100==0):

143.         if(year%400==0):

144.             print("It is a leap year")

145.         else:

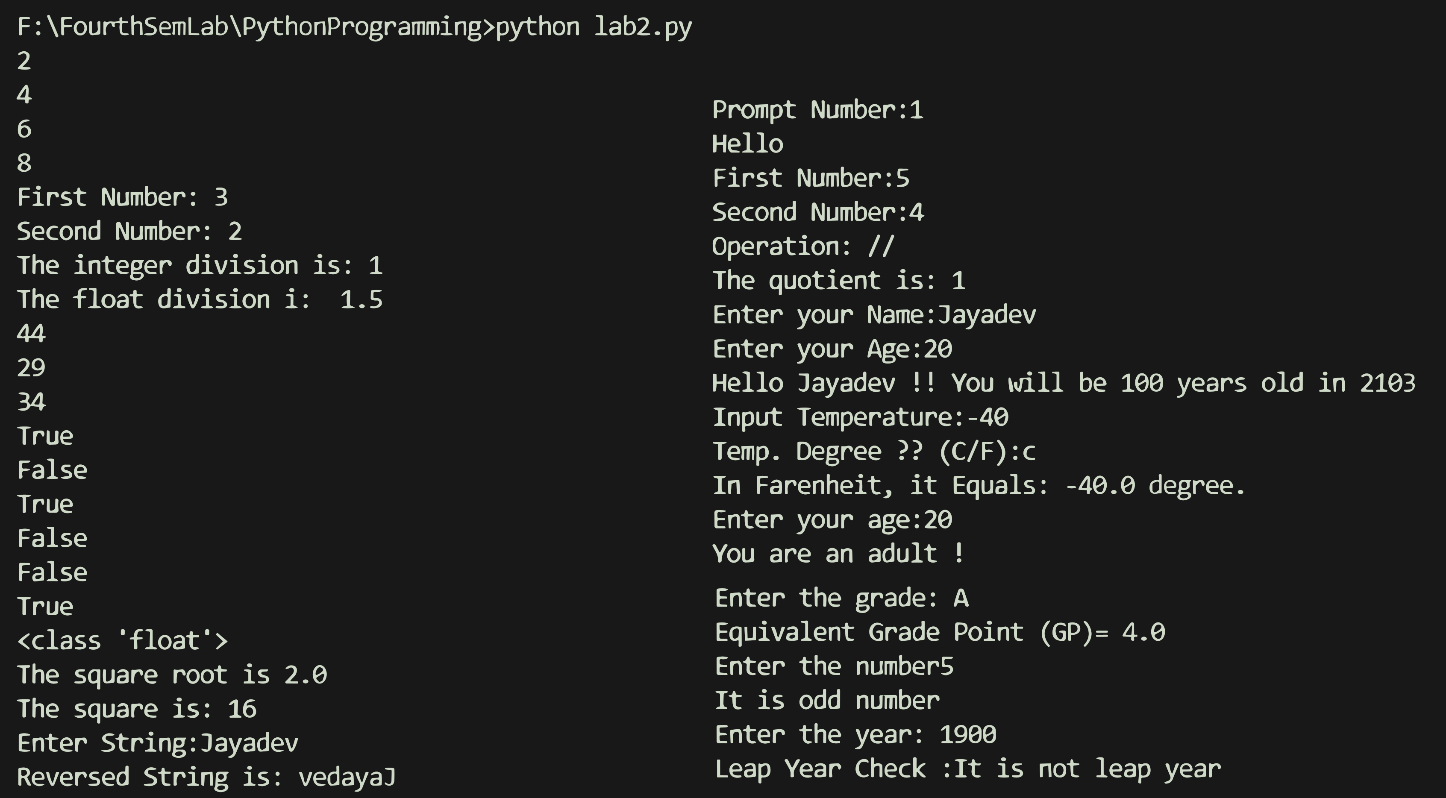
146.             print("It is not leap year")

147.     else:

148.         print("It is a leap Year")

149. else:

150.     print("It is not leap Year")



**Conslusion**:

Conditional statements and branching were implemented in the python programming with different forms of if, elif and else conditions and nested conditions too. Match-case was also used for branching the program to different states upon certain conditions. The lab program and the outcome of the program are as shown above. These result clarifies the understanding of conditional statements and branching with python programming.