Programming Essentials in Python

2025-2026

Questions

Write a program that ask user to enter two integer number and then print the first number divided by the second number?

```
1 num1=eval(input("Enter the First Numer: "))
2 num2=eval(input("Enter the Second Numer: "))
3 result=num1/num2
4 print("The Result is",result)
```

What will happen if the user enter the second number zero?

Course Contents

- Introduction to programming, basic data types and operators.
- Strings, Lists, Tuples, and Dictionaries.
- **Control Statements.**
- Functions.
- Object-oriented concepts and their implementations.

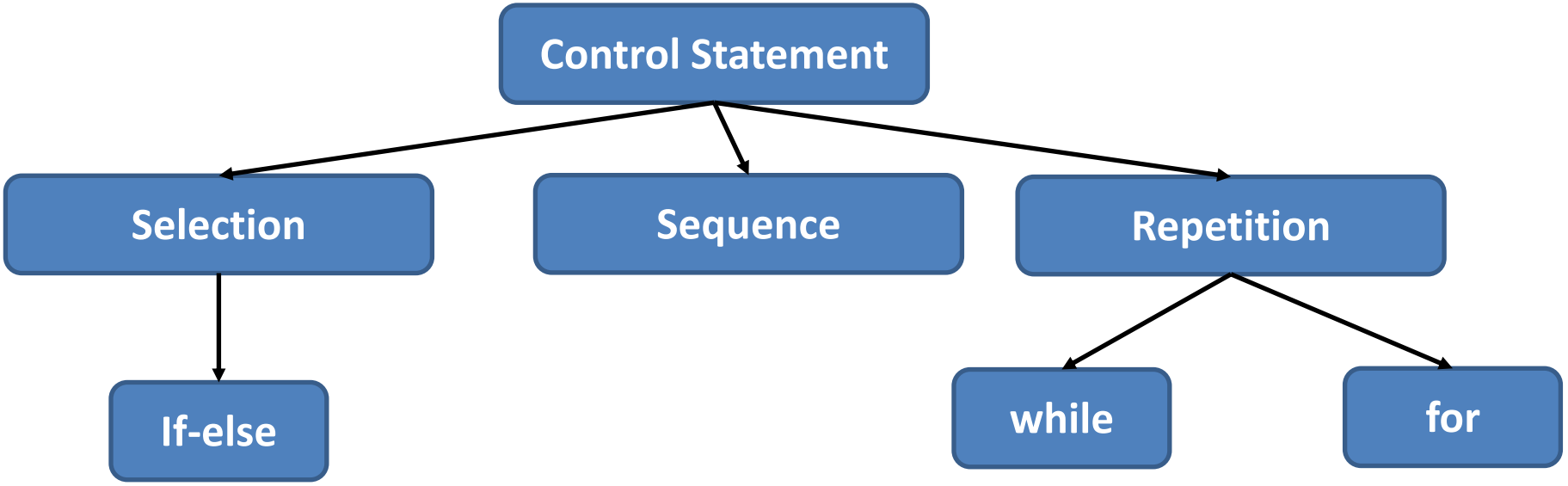
Control Statements

Outline

- Type of Control Statements.
- If and If else statement with examples.
- Nested If with examples.

Type of Control Statements.

There are three types of Control Statements are given as follows:



- **Sequence:** execute statements one after the other.
- **Selection**: select which block statement(s) to be executed.
- **Repetition:** repeatedly execute a block of statement(s).

Relational Operators

Relational Operators are used in selection and repetition statements to compare numbers, characters and strings and determine the execution order.

Operator	Means
==	Equal To
!=	Not Equal To
<	Less Than
<=	Less Than or Equal To
>	Greater Than
>=	Greater Than or Equal To

Relational Expressions

Relational Expressions: The result of Relation expression can be a Boolean value – true or false

Examples:

12 > 5 is true

7 <= 5 is false

```
int i = 5;
int k = 12;

p = (i < 10);  ➔ True
q = (k < i);   ➔ False
r = (i >= k);  ➔ False
s = (k <= 12); ➔ True
```

Selection Statement

Selection statements are used to choose among alternative courses of action.

Python provides the following selection statements:

- if statementif-
- else statementif-
- elif statement
- nested if statement

Selection Statement: If Statement

A program can decide which statements to execute based on an expression

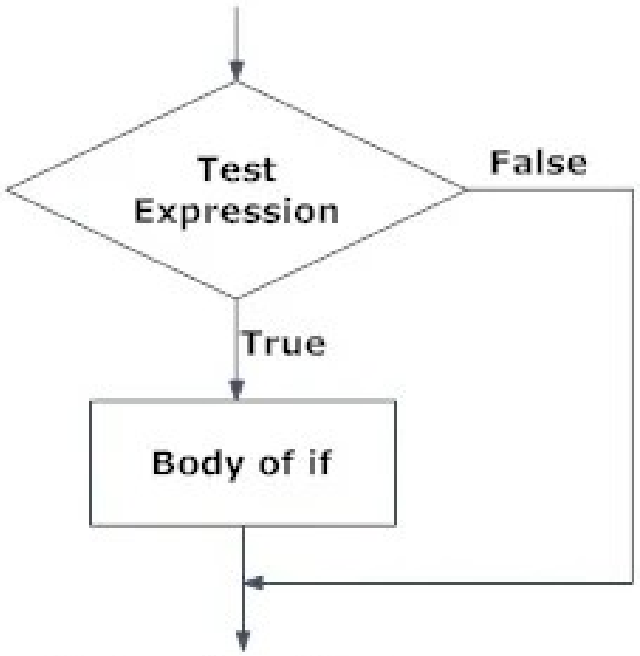


Fig: Operation of if statement

Syntax:

```
if (Expression):
    action statement
```

In Python, the body of the if statement is indicated by the indentation.

Evaluation:

- ✓ **If (expression) is true, then statement is executed.**
- ✓ **If (expression) is false, then statement is skipped – not executed**

Selection Statement: If Statement

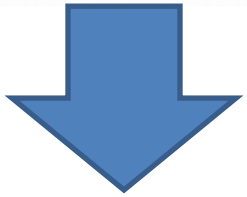
Write a program that ask user to enter two integer number and then print the first number divided by the second number?

```
1 num1=eval(input("Enter the First Numer: "))
2 num2=eval(input("Enter the Second Numer: "))
3 if(num2!=0):
4     result=num1/num2
5     print("The Result is",result)
```

Selection Statement: If Statement

What will happen if you remove the Indentation from the if statement?

```
1 num1=eval(input("Enter the First Number: "))
2 num2=eval(input("Enter the Second Number: "))
3 if(num2!=0):
4 result=num1/num2
5 print("The result is", result)
```



```
>>> %Run Test.py
Traceback (most recent call last):
  File "C:\Users\CM\Desktop\Test.py", line 4
    result=num1/num2
    ^^^^^^
IndentationError: expected an indented block after 'if'
statement on line 3
>>>
```

Selection Statement: If Statement

Write a program that accepts a radius from the user and in case this radius is greater than or equal to 0, then calculate the area and print it.

```
1 radius=eval(input("Enter the Radius: "))
2 if(radius>=0):
3     area=radius*radius*3.14
4     print("The Area is",area)
```

Selection Statement: If Statement

Write a program that accept an integer number from the user and in case this number is even print out the following message:

"This number is even".

```
1 number=eval(input("Enter the Number: "))
2 if(number%2==0):
3     print("This number is even")
```

Selection Statement: If-else Statement

Allows choice between statements if (expression) is true or false.

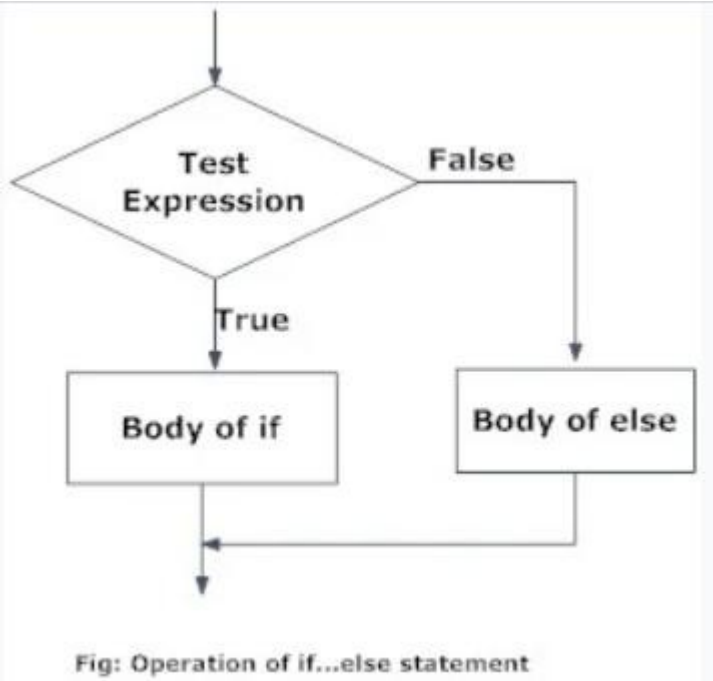


Fig: Operation of if...else statement

Syntax:

```
if (Expression):
    action statement1
else:
    action statement2
```

Evaluation:

- ✓ **If (expression) is true, then statement1 is executed.**
- ✓ **If (expression) is false, then statement2 is executed**

Selection Statement: If Statement

Write a program that accepts an integer number from the user and prints whatever this number is even or odd.

```
1 number=eval(input("Enter the Number: "))
2 if(number%2==0):
3     print("This number is even")
4 else:
5     print("This number is odd")
```

Selection Statement: If-else Statement

Write a program that asks the user to enter an integer number and then print if the number is positive or negative.

```
1 number=eval(input("Enter the Number: "))
2 if(number>=0):
3     print("This number is Postive")
4 else:
5     print("This number is Negative")
```

Selection Statement: If-else Statement

Write a program that accepts two integers from the user and then prints which one is greater than the other..

```
1 num1=eval(input("Enter the First Number: "))
2 num2=eval(input("Enter the Second Number: "))
3 if(num1>num2):
4     print(num1, "is greater than", num2)
5 else:
6     print(num2, "is greater than", num1)
```

Selection Statement: If-else Statement

Logical Operators are used in selection and repetition statements to combine more than one expression.

Operator	Description	Example
and	Equal To	X>5 and x<10
or	Not Equal To	X>5 or x<4
not	Less Than	not(x<5)

```
1 num1=eval(input("Enter the First Number: "))
2 num2=eval(input("Enter the Second Number: "))
3 if(num1>=0 and num2>=0):
4     print("Both numbers are Postive")
```

Selection Statement: If-elif…-else Statement

Allows testing multiple conditions.

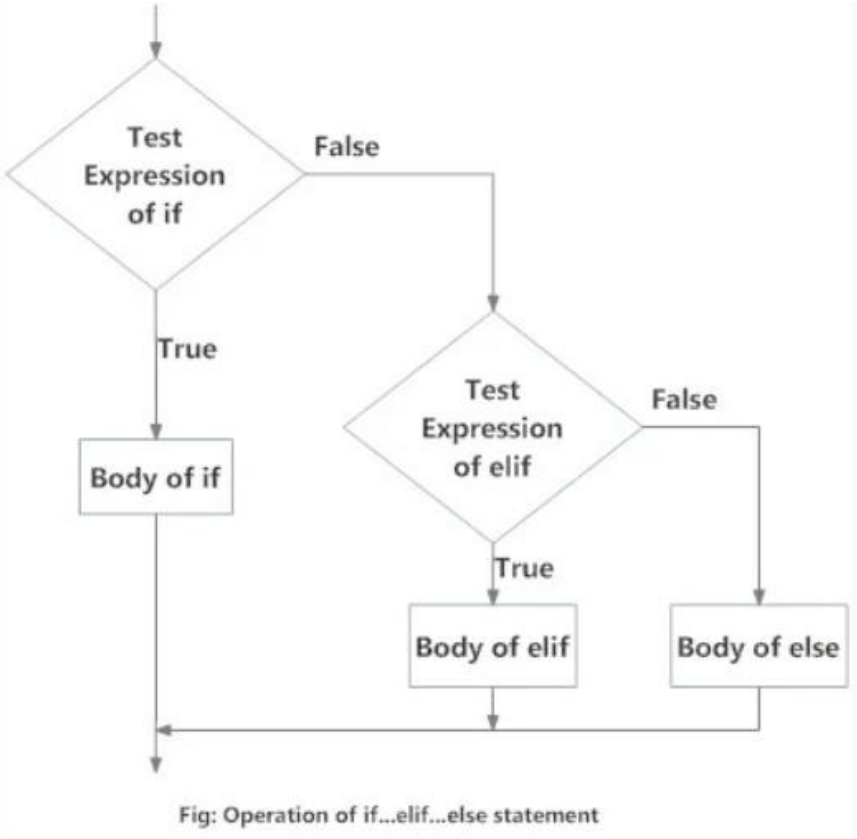


Fig: Operation of if...elif...else statement

Syntax:

```
if (Expression1):
action statement1
elif (Expression2):
    action statement2
else:
    action statement3
```

Evaluation:

- ✓ **If (expression1) is true, then statement1 is executed.**
- ✓ **If (expression2) is true, then statement2 is executed.**
- ✓ **Else, then statement3 is executed**

Selection Statement: If-elif…-else Statement

Write a program that accepts the score of the student and print A if it is greater than or equal to 90, print B if it is greater than or equal to 80, print C if it is greater than or equal to 70, print D if it greater than or equal 60, otherwise, print F.

Selection Statement: If-elif…-else Statement

```
1  score=eval(input("Enter the Score: "))
2  if(score>=90):
3      print("A")
4  elif(score>=80):
5      print("B")
6  elif(score>=70):
7      print("C")
8  elif(score>=60):
9      print("D")
10 else:
11     print("F")
```

Selection Statement: Nested If Statement

Nested If ➔ Means to write an if statement within another if statement

Syntax:

```
if ( Expression):
    if ( Expression)
        action statement1
    else:
        action statement2
else:
    action statement3
```

Selection Statement: Nested If Statement

```
1 number=eval(input("Enter the number: "))
2 if(number>=0):
3     if(number==0):
4         print("Zero")
5     else:
6         print("Postive")
7 else:
8     print("Negative")
```

Output 1

```
Enter the number: 4
Postive
```

Output 2

```
Enter the number: 0
Zero
```

Output 3

```
Enter the number: -3
Negative
```

Selection Statement: Nested If Statement

Write a program that accepts an integer number from the user, in case the number is Positive, check and print out whether it is an Even or Odd number.

```
1 number=eval(input("Enter the number: "))
2 if(number>=0):
3     if(number%2==0):
4         print("Even number")
5     else:
6         print("Odd Numer")
7 else:
8     print("Error, Negative Number")
```

THANKS

for your attention