Practical 13

Aim: Write a function that asks for an integer and prints square of it. Use a while loop with a try, except, else block to account for incorrect inputs.

Theory:

In Python, the `try-except-else` block is used to handle exceptions in the code. The basic idea is to try to execute a block of code and if an exception occurs during the execution, catch and handle it using the `except` block. If no exception occurs, the code in the `else` block is executed.

Here's an example of using `try-except-else` block in Python:

```
try:
    num = int(input("Enter a number: "))
    result = 100 / num
except ZeroDivisionError:
    print("Cannot divide by zero")
except ValueError:
    print("Invalid input")
else:
    print("Result:", result)
```

In the above code, we're trying to take user input as an integer and dividing 100 by it. If the user enters a non-zero number, the division will be performed, and the result will be printed. However, if the user enters zero or a non-numeric value, an exception will occur, and the appropriate message will be printed.

`ValueError` is a built-in exception in Python that is raised when the input provided by the user cannot be converted into the expected data type. In the above example, we're catching `ValueError` exception in the `except` block and printing an error message when the user enters a non-numeric value.