4.1.1 Exception Handling

What is Exception Handling?

Exception handling is the process of detecting and responding to runtime errors (exceptions) so that a program can continue running or terminate gracefully instead of crashing.

How exception handling works

Most programming languages use three main keywords:

- **try** – Contains the code that might cause an exception.
- **except** (or **catch** in some languages) – Runs if an exception occurs.
- **finally** – Runs whether or not an exception occurs. It is often used to close files or release resources.

Why is exception handling important?

Without exception handling:

- The program may stop unexpectedly (crash).
- The user could lose unsaved work.
- The program might display confusing error messages.

With exception handling:

- The program can display a helpful message to the user.
- The program can recover from the error and continue running where appropriate.
- The programmer can log the error for debugging.

Common exceptions

Some common runtime exceptions include:

- **Division by zero** – trying to divide a number by 0.
- **File not found** – attempting to open a file that does not exist.
- **Invalid input** – the user enters text when a number is expected.
- **Index out of range** – trying to access an item that does not exist in a list or array.

Python Example

```python
try:
    number = int(input("Enter a number: "))
    result = 100 / number
    print(result)

except ZeroDivisionError:
    print("You cannot divide by zero.")

except ValueError:
    print("Please enter a valid number.")

finally:
    print("Program finished.")
```