• Assertions
  - Statements used to check the correctness of assumptions made in a program during run-time.
  - Debug tool
  - C example:

    ```c
    void assert (int expression);
    
    • If the expression meaning is 0 (false), the expression, source code filename, and line number are sent to standard error, and then the abort function is invoked.
    
    // GPACalc.c
    #include <stdio.h>
    #include <assert.h>
    float GPACalc (int size, float *ary) {
      float sum = 0;
      for (int i = 0; i < size; i++) {
        // check invalid gpa
        assert (ary[i] >= 0.0 && ary[i] <= 4.3);
        sum += ary[i];
      }
      return sum/size;
    }
    
    int main () {
      float a[] = {3.5, 3.8, 4.0, 4.1, 4.5};
      printf("%f\n", GPACalc(5, a));
      return 0;
    }
    ```

  - Java example:
    ```java
    assert expression;
    assert expression1 : expression2;
    
    • expression1: boolean expression; If false, JVM throws AssertionError.
    • expression2: an expression that has a value.
    • java -ea: enable assertions
- **Python example**

```python
assert condition, error_message(optional)
```

- condition: boolean expression
- error_message: printed on terminal if AssertionError

```python
# GPACalc.py
def GPACalc (ary):
    sum = 0.0
    for f in ary:
        assert f >= 0.0 and f <= 4.3, "Invalid value " + str(f)
        sum += f;
    return sum/len(ary)

def main ():
    a = [3.5, 3.8, 4.0, 4.1, 4.5]
    print(GPACalc(a))

if __name__ == "__main__":
    main()
```