Mohammad Aship

L3C9, Workshop-4

1 Task – 6

- Why the output is incorrect?

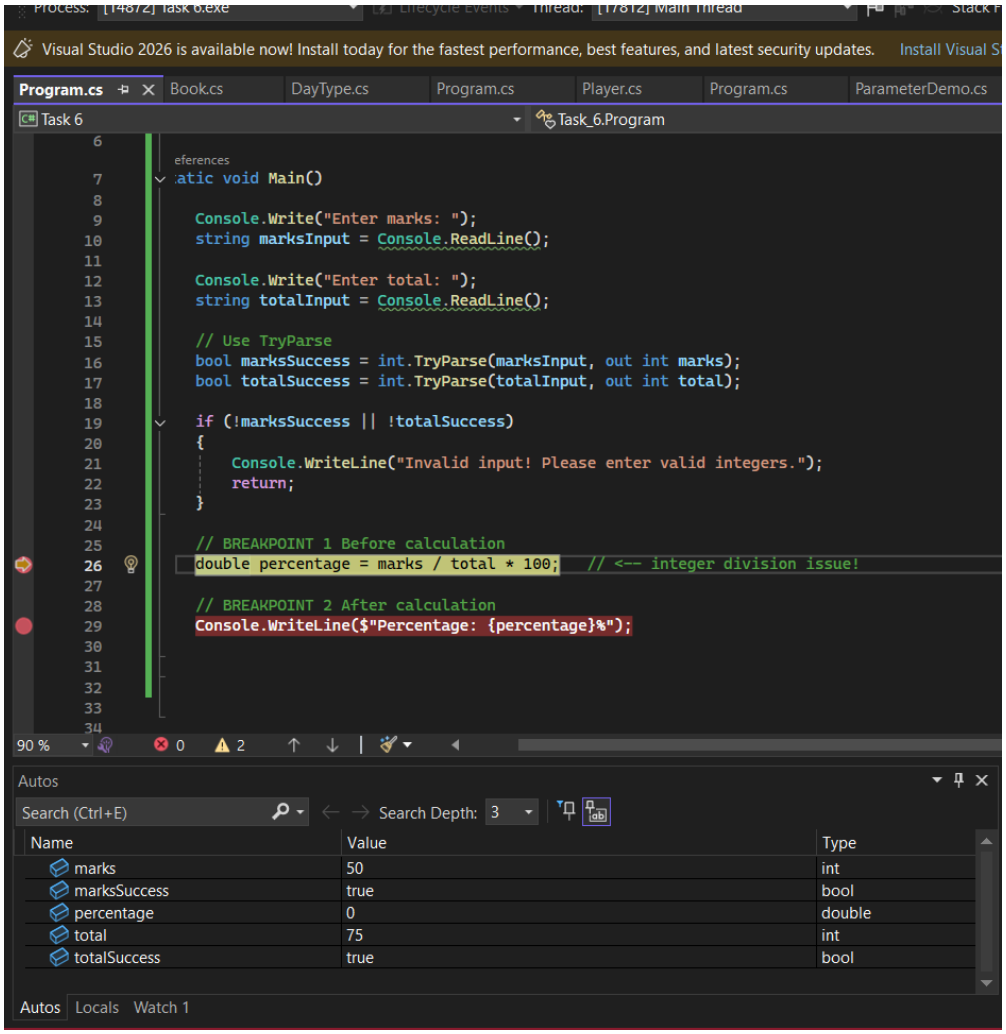

Figure 1 Integer division error.

Since the output which is percentage is shown Zero but it should contain some value since the marks and total was given 50/75 as an input. It has given error output because the total is being

multiplied by 100 which should be done directly and the marks and total are integer which are performing integer division but the percentage is in double which shows directly zero. For the correct answer marks/total should be surrounded by parenthesis and that whole should get multiplied by 100 which can show the correct answer.

- How can we correct the program?

  We can correct the program by changing or parsing the integer division value of marks and total to double which matches and become compatible with percentage datatype and shows the correct and exact answer of marks in percentage.

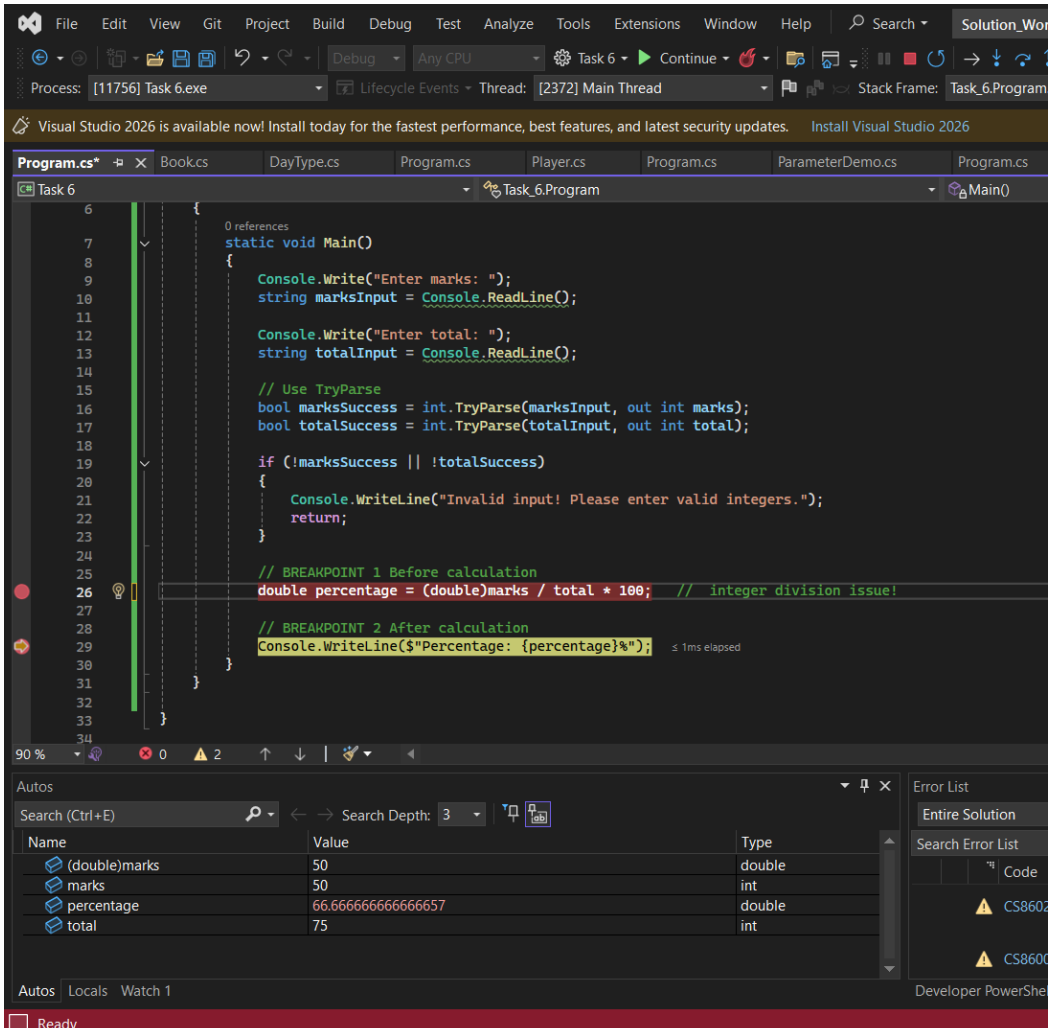

Figure 2 Corrected Program

2 Task – 7

- How constructors help in software development?

Since the constructor are made in a class where the fields and methods(behaviors) are included and to perform a huge task to perform a code as a reusability. Like to perform a library and its books information and details store in a system. By using the constructor where the class have different unique needed fields for books information store different details of a different books by creating an object of a particular class and its constructor. (Oracle, 2024)

Three Real-World Example:

- **Database Connection Setup**
  When creating a database service class, constructors inject the connection string and initialize the connection object.

- **Management System**
  To store the huge number or frequency of a data of a same category by avoiding useless code and increase the code reusability. Constructor of a class is made and through object it gets the connected to take value in a parameter and save as a object.

- **API Client Initialization**

  Rest API clients often need authorization keys and base URLs. Constructors ensure these are always set. This constructor helps in preventing API calls without authentication. (Microsoft, 2025)

- **OOP Principle:**

- **What is Encapsulation?**

  Encapsulation is one of the core principles of Object-Oriented Programming (OOP). It refers to bundling data (variables) and methods (functions) together into a single unit (class) and restricting direct access to some components of that object.

  Encapsulation ensures that:

  - The state of an object can only be modified through well-defined interfaces**.**
  - The internal representation of data remains opaque to the user.
  - The design promotes modularity, data integrity, and abstraction of implementation details**.**

- **Classes**

A class is considered a formal specification or template that defines the structure and behavior of objects.
From a theoretical standpoint, a class describes:

- Attributes (state variables)
- Methods (operations or behavior)

In software design theory, classes are the conceptual tools used to define real-world entities in terms of properties and operations, forming the basis for modeling complex systems.

- **Objects**

An object is an **instance** of a class and represents a concrete element created in memory based on the class definition.
In OOP theory, an object encapsulates:

- A **state**, representing specific values stored at a particular time
- A **behavior**, defined by the methods it can perform

Objects interact with each other through **message passing**, which promotes modularity and distributed responsibilities in the system.

- **Two Examples of how we can use OOPs Principles:**

    - Information Hiding:
      Encapsulation enforces information hiding those internal details of a module or object must be hidden and the essential information should be exposed only only to the outside world not the complex one which is define by encapsulation.

    - Data Integrity and Controlled Access:
      Encapsulation also prevents unauthorized or unintended modification of to data. In Encapsulation by exposing only specific methods internal state can remain valid and consistent. And also, the class can enforce predetermined rules, ensuring system reliability. (Microsoft, 2025)

3 References

Microsoft. (2025) *Learn* [Online]. Available from: https://learn.microsoft.com/en-us/dotnet/csharp/programming-guide/classes-and-structs/constructors [Accessed 20 November 2025].

Oracle. (2024) *Java Docuentation* [Online]. Available from: https://docs.oracle.com/javase/tutorial/java/javaOO/constructors.html [Accessed 20 November 2025].