

DCIT 318 Assignment 1 Documentation

Programming II - Console Applications

Student Name: Samuel Gyasi Fordjour
Student ID: 11053488
Course: DCIT 318 - Programming II
Assignment: Assignment 1 - Console Applications

Table of Contents

- 1. [Overview](#)
- 2. [System Requirements](#)
- 3. [Installation and Setup](#)
- 4. [Compilation Instructions](#)
- 5. [Application 1: Grade Calculator](#)
- 6. [Application 2: Ticket Price Calculator](#)
- 7. [Application 3: Triangle Type Identifier](#)
- 8. [Troubleshooting](#)
- 9. [Technical Features](#)
- 10. [Conclusion](#)

Overview

This assignment consists of three console applications written in C#:

- 1. **Grade Calculator** - Converts numerical grades to letter grades with performance descriptions
- 2. **Ticket Price Calculator** - Calculates movie ticket prices based on age with discount application
- 3. **Triangle Type Identifier** - Identifies triangle types and calculates geometric properties

All applications feature:

- Menu-driven interfaces
- Comprehensive error handling
- Input validation
- User-friendly navigation
- Professional output formatting

System Requirements

Minimum Requirements:

- **Operating System:** Windows 7 or later
- **.NET Framework:** 4.0 or later
- **Memory:** 512 MB RAM
- **Storage:** 50 MB available space

Recommended:

- **Operating System:** Windows 10/11
- **.NET Framework:** 4.8
- **Memory:** 2 GB RAM
- **IDE:** Visual Studio 2019/2022 or Visual Studio Code

Installation and Setup

Method 1: Using .NET Framework Compiler

- 1. **Locate the C# Compiler:**

```
C:\Windows\Microsoft.NET\Framework64\v4.0.30319\csc.exe
```

- 2. **Download Source Files:**

- GradeCalculator.cs

- TicketPriceCalculator.cs
- TriangleTypeIdentifier.cs

3. Create Project Directory:

```
mkdir DCIT318Assignment1
cd DCIT318Assignment1
```

Method 2: Using Visual Studio

1. Open Visual Studio
2. Create new Console Application project
3. Copy source code into Program.cs
4. Build and run

Compilation Instructions

Command Line Compilation

Open Command Prompt or PowerShell and navigate to the project directory:

```
# Navigate to project directory
cd "d:\S.K\LECTURES\L300 2nd Sem\PROGRAMMING II\Assignments\dcit318-assignment1-11053488"

# Compile Grade Calculator
C:\Windows\Microsoft.NET\Framework64\v4.0.30319\csc.exe GradeCalculator.cs

# Compile Ticket Price Calculator
C:\Windows\Microsoft.NET\Framework64\v4.0.30319\csc.exe TicketPriceCalculator.cs

# Compile Triangle Type Identifier
C:\Windows\Microsoft.NET\Framework64\v4.0.30319\csc.exe TriangleTypeIdentifier.cs
```

Running the Applications

After successful compilation:

```
# Run Grade Calculator
.\GradeCalculator.exe

# Run Ticket Price Calculator
.\TicketPriceCalculator.exe

# Run Triangle Type Identifier
.\TriangleTypeIdentifier.exe
```

Visual Studio Compilation

1. Open Visual Studio
2. File → New → Project → Console Application
3. Copy application source code
4. Press F5 to build and run

Application 1: Grade Calculator

Purpose

Converts numerical grades (0-100) to letter grades with performance descriptions and grading scale information.

Features

- **Grade Conversion:** A, B, C, D, F grading system
- **Performance Descriptions:** Excellent, Good, Average, Below Average, Fail

- **Grading Scale Display:** Complete scale with ranges
- **Multiple Calculations:** Process multiple grades in one session

Main Menu Options

```
===== MAIN MENU =====  
Please select an option:  
1. Calculate Grade  
2. View Grade Scale  
3. Exit Application  
=====
```

How to Use

Step 1: Start the Application

- Run `GradeCalculator.exe`
- Welcome screen will display

Step 2: Navigate Main Menu

- Enter `1` to calculate grades
- Enter `2` to view grading scale
- Enter `3` to exit

Step 3: Calculate Grades

1. Select option `1` from main menu
2. Enter numerical grade (0-100)
3. View results:
 - Numerical grade
 - Letter grade
 - Performance description
4. Choose to calculate another grade or return to menu

Step 4: View Grading Scale

1. Select option `2` from main menu
2. Review complete grading criteria:
 - A: 90-100 (Excellent)
 - B: 80-89 (Good)
 - C: 70-79 (Average)
 - D: 60-69 (Below Average)
 - F: 0-59 (Fail)
3. Press any key to return to menu

Input Validation

- **Range Check:** Grades must be 0-100
- **Format Check:** Only numerical input accepted
- **Empty Input:** Prevents crashes from empty entries
- **Special Values:** Handles NaN and Infinity

Sample Usage

```
Enter a numerical grade between 0 and 100:  
85  
Grade: 85  
Letter Grade: B  
Performance: Good  
  
Do you want to calculate another grade? (y/n): n
```

Application 2: Ticket Price Calculator

Purpose

Calculates movie ticket prices based on customer age with automatic discount application for children and seniors.

Pricing Structure

- **Regular Price:** GHC 10.00 (Ages 13-64)
- **Child Discount:** GHC 7.00 (Ages 0-12)
- **Senior Discount:** GHC 7.00 (Ages 65+)

Features

- **Age-Based Pricing:** Automatic discount calculation
- **Savings Display:** Shows money saved with discounts
- **Pricing Information:** Complete pricing structure display
- **Multiple Calculations:** Process multiple customers

Main Menu Options

```
===== MAIN MENU =====
Please select an option:
1. Calculate Ticket Price
2. View Pricing Information
3. Exit Application
=====
```

How to Use

Step 1: Start the Application

- Run `TicketPriceCalculator.exe`
- Welcome screen will display

Step 2: Navigate Main Menu

- Enter `1` to calculate ticket prices
- Enter `2` to view pricing information
- Enter `3` to exit

Step 3: Calculate Ticket Price

1. Select option `1` from main menu
2. Enter customer age
3. View results:
 - Age entered
 - Ticket price with category
 - Discount message
 - Savings amount (if applicable)
4. Choose to calculate another price or return to menu

Step 4: View Pricing Information

1. Select option `2` from main menu
2. Review complete pricing structure:
 - Regular and discount prices
 - Age categories
 - Savings amounts
3. Press any key to return to menu

Input Validation

- **Age Range:** 0-150 years accepted
- **Negative Values:** Prevents negative ages
- **Format Check:** Only numerical input accepted
- **Realistic Ages:** Warns for unrealistic values

Sample Usage

```
Please enter your age:
8
Age: 8
Ticket Price: GHC7.00 (Child Discount)
You qualify for the child discount!
You save: GHC3.00 compared to regular price!
```

Application 3: Triangle Type Identifier

Purpose

Identifies triangle types based on side lengths and calculates geometric properties including perimeter and area.

Triangle Types

- **Equilateral:** All three sides equal
- **Isosceles:** Two sides equal
- **Scalene:** No sides equal

Features

- **Triangle Classification:** Automatic type identification
- **Validation:** Triangle inequality theorem checking
- **Property Calculation:** Perimeter and area using Heron's formula
- **Educational Content:** Triangle information and examples

Main Menu Options

```
===== MAIN MENU =====
Please select an option:
1. Identify Triangle Type
2. View Triangle Information
3. Exit Application
=====
```

How to Use

Step 1: Start the Application

- Run TriangleTypeIdentifier.exe
- Welcome screen will display

Step 2: Navigate Main Menu

- Enter 1 to identify triangle types
- Enter 2 to view triangle information
- Enter 3 to exit

Step 3: Identify Triangle Type

1. Select option 1 from main menu
2. Enter three side lengths:
 - First side length
 - Second side length
 - Third side length
3. View results:
 - Sides entered
 - Triangle type classification
 - Detailed explanation
 - Perimeter calculation
 - Area calculation
4. Choose to analyze another triangle or return to menu

Step 4: View Triangle Information

1. Select option 2 from main menu
2. Review educational content:
 - Triangle type definitions

- Characteristics of each type
 - Example measurements
 - Triangle inequality rule
3. Press any key to return to menu

Input Validation

- **Positive Values:** All sides must be positive
- **Triangle Inequality:** Validates mathematical possibility
- **Format Check:** Only numerical input accepted
- **Special Values:** Handles NaN and Infinity

Mathematical Formulas

- **Perimeter:** $P = \text{side1} + \text{side2} + \text{side3}$
- **Area (Heron's Formula):**
 - $s = (\text{side1} + \text{side2} + \text{side3}) / 2$
 - $\text{Area} = \sqrt{s \times (s - \text{side1}) \times (s - \text{side2}) \times (s - \text{side3})}$

Sample Usage

```
Enter the length of the first side: 3
Enter the length of the second side: 4
Enter the length of the third side: 5
```

```
Sides entered: 3, 4, 5
Triangle Type: Scalene
No sides are equal.
```

```
Triangle properties:
- Perimeter: 12.00
- Area: 6.00 square units
```

Troubleshooting

Common Issues and Solutions

Compilation Errors

Problem: "csc is not recognized"

```
Solution: Use full path to compiler:
C:\Windows\Microsoft.NET\Framework64\v4.0.30319\csc.exe filename.cs
```

Problem: "String interpolation not supported"

```
Solution: Applications already use C# 5 compatible syntax
```

Problem: "Access denied"

```
Solution: Run command prompt as administrator
```

Runtime Errors

Problem: Application crashes on input

```
Solution: Applications include comprehensive error handling
Enter valid numerical values as prompted
```

Problem: Unexpected behavior

```
Solution: Follow menu prompts exactly
Use only specified input formats
```

Input Issues

Problem: Invalid grade ranges

Solution: Enter grades between 0-100 for Grade Calculator

Problem: Invalid triangle sides

Solution: Ensure sides are positive and satisfy triangle inequality

Problem: Invalid age values

Solution: Enter realistic ages (0-150) for Ticket Calculator

Getting Help

1. **Read Error Messages:** Applications provide detailed error descriptions
2. **Check Input Format:** Ensure numerical inputs are properly formatted
3. **Follow Menu Options:** Use only listed menu choices
4. **Restart Application:** If persistent issues occur

Technical Features

Error Handling

- **Exception Types Handled:**
 - `FormatException` (invalid number formats)
 - `OverflowException` (numbers too large/small)
 - `ArgumentException` (invalid arguments)
 - `OutOfMemoryException` (system memory issues)
 - `StackOverflowException` (stack overflow protection)

Input Validation

- **Null/Empty Checks:** Prevents crashes from empty input
- **Range Validation:** Ensures input within acceptable ranges
- **Format Validation:** Confirms numerical input where required
- **Special Value Handling:** Manages NaN and Infinity values

User Experience

- **Menu-Driven Interface:** Easy navigation between features
- **Clear Instructions:** Step-by-step guidance
- **Professional Output:** Formatted results with proper spacing
- **Graceful Recovery:** Continues operation after errors

Code Architecture

- **Modular Design:** Separated functions for different features
- **Comprehensive Documentation:** XML documentation for all methods
- **Clean Code:** Readable and maintainable structure
- **Best Practices:** Following C# coding standards

Conclusion

This assignment demonstrates proficiency in:

1. **Console Application Development:** Creating user-friendly command-line applications
2. **Error Handling:** Implementing robust exception management
3. **Input Validation:** Ensuring data integrity and user safety
4. **Menu Systems:** Designing intuitive navigation interfaces
5. **Mathematical Computations:** Applying formulas for practical calculations
6. **Code Documentation:** Providing comprehensive technical documentation

Learning Outcomes Achieved:

- Understanding of C# console application structure
- Implementation of user input validation
- Application of mathematical concepts in programming

- Development of professional user interfaces
- Creation of maintainable and documented code

Future Enhancements:

- Graphical user interface implementation
 - Database integration for data persistence
 - Advanced mathematical calculations
 - Multi-language support
 - Configuration file management
-

End of Documentation

This document serves as a complete guide for understanding, compiling, and using the DCIT 318 Assignment 1 console applications. For additional support or questions, please refer to the troubleshooting section or contact the course instructor.