C# Sample
By Jafir Islam Siam

Quiz-1 syllabus :

Output Tracing

Show output of the following codes

#1.

```
int a = 10;
int b = 5;
double c = 4.75;
string str = "25";
bool condition1 = true;
bool condition2 = false;

int result = a + b;
result *= b;
result -= a;
double divisionResult = result / c;

int parsedInt = int.Parse(str);
int roundedC = (int)c;

bool logicAndComparison = (a > b) && (parsedInt == a);
bool equalityCheck = divisionResult == (double)a / b;

Console.WriteLine("Final result after arithmetic: " + result);
Console.WriteLine("Division result: " + divisionResult);
Console.WriteLine("Parsed integer: " + parsedInt);
Console.WriteLine("Rounded double to int: " + roundedC);
Console.WriteLine("Logic and comparison result: " + logicAndComparison);
Console.WriteLine("Equality check for division result: " + equalityCheck);
```

#2.

```
int addResult = x + y;
int modResult = x % y;
bool conditionResult = (addResult > modResult) && flag;

double castedResult = (double)(addResult * modResult) / z;

int intFromDouble = (int)z;
string strNum = "100";
int parsedStrNum = int.Parse(strNum);

Console.WriteLine("Addition result: " + addResult);
Console.WriteLine("Modulus result: " + modResult);
Console.WriteLine("Condition result: " + conditionResult);
Console.WriteLine("Casted double result: " + castedResult);
Console.WriteLine("Parsed integer from string: " + parsedStrNum);
Console.WriteLine("Int value from double: " + intFromDouble);
```

#3.

```
double div = a / (double)b;
Console.WriteLine("Division result (as double): " + div);

int intResult = (int)(num * div);
Console.WriteLine("Integer result from casting: " + intResult);

bool logicResult = check1 || (check2 && (a > b));
Console.WriteLine("Logical OR/AND combination result: " + logicResult);

int parsedStr = int.Parse(strNum);
int finalResult = parsedStr - intResult;
Console.WriteLine("Final result after subtraction: " + finalResult);
```

#4.

```
double sum = x + y;
Console.WriteLine("Sum of x and y: " + sum);

int parsedStr = int.Parse(str);
bool comparisonResult = (parsedStr == x) && (y > x);
Console.WriteLine("Comparison with parsedStr and y: " + comparisonResult);

double modAndDiv = parsedStr % x + (x / y);
Console.WriteLine("Modulus and division result: " + modAndDiv);

condition = (sum > parsedStr) || condition;
Console.WriteLine("Updated condition: " + condition);
```

#5.

```
int multResult = a * b;
Console.WriteLine("Multiplication result: " + multResult);

double divResult = multResult / num;
Console.WriteLine("Division result (int * double): " + divResult);

bool finalCondition = (a < b) || (flag1 && !flag2);
Console.WriteLine("Logical result of conditions: " + finalCondition);

int parsedInt = int.Parse(strVal);
Console.WriteLine("Parsed integer: " + parsedInt);

int modResult = multResult % parsedInt;
Console.WriteLine("Modulus result with parsed int: " + modResult);

double sumResult = divResult + parsedInt;
Console.WriteLine("Sum of divResult and parsedInt: " + sumResult);
```

Code Writing

1.Question: Arithmetic, Assignment, and Logical Operations for Store Discounts

Write a program to calculate a store discount based on two variables: purchaseAmount (integer) and discount (integer representing the discount percentage). The program should perform the following:

- Calculate the total discount amount by multiplying purchaseAmount by discount and dividing by 100. Print the discount amount.
- Use assignment operators to decrease purchaseAmount by the discount amount and print the final payable amount.
- Check if the payable amount is over 100; if so, display "Large Purchase," otherwise display "Standard Purchase."
- Use logical operators to check if both the discount percentage is greater than 10 and the payable amount is less than 80. Print "Special Offer" if true; otherwise, print "Regular Offer."

Ensure each calculation and conditional check has clear output labels.

2.Multi-Step Type Casting and Condition Checking for School Grading System

Create a program that assigns scores to students based on their performance. The program should do the following:

- Take three inputs: a double averageScore, an integer projectsCompleted, and a string bonusPoints that can be converted to an integer.
- Convert bonusPoints to an integer and add it to projectsCompleted.
- Multiply this sum by averageScore, then cast it to an integer and store it as finalGrade.
- If finalGrade is 90 or above, display "Outstanding"; if between 70 and 89, display "Satisfactory"; otherwise, display "Needs Improvement."
- Additionally, check if either projectsCompleted is greater than 5 or finalGrade is greater than or equal to 80. Print "Eligible for Honors" if true; otherwise, "No Honors."

Be sure to print each conversion and calculation result with descriptive labels.

3. Question: Type Casting and Parsing for Loan Eligibility

Create a program to determine loan eligibility based on user inputs. The program should prompt the user for three inputs: salary (double), yearsOfExperience (integer), and creditScore (string representing an integer). Then:

- Parse creditScore into an integer.
- If salary multiplied by yearsOfExperience is greater than 100,000, cast this product to an integer and display it as the "Eligibility Score."
- Check if Eligibility Score is greater than creditScore. If true, print "Eligible for High Loan," otherwise, "Eligible for Standard Loan."
- Also, display a final message: if both salary is over 50,000 and yearsOfExperience is over 5, print "Premium Applicant"; otherwise, print "Standard Applicant."

Make sure each result is labeled clearly.

4. Question: Complex Condition Evaluation in Finance Calculation

Write a program that calculates the eligibility for a financial incentive based on a person's performance. The program should:

- Initialize monthlyIncome as a double, yearsAtCompany as an integer, and rating as a string representing a double.
- Parse rating to a double and calculate performanceScore by multiplying monthlyIncome by rating.
- Use explicit casting to convert performanceScore to an integer and store it as roundedScore.
- Check if roundedScore divided by yearsAtCompany has a remainder of 0. If true, print "Consistent Performance," otherwise "Inconsistent Performance."
- Lastly, use logical operators to check if both monthlyIncome is over 5000 and rating is above 3.0. Print "Eligible for Incentive" if true; otherwise, print "No Incentive."

Each output should include clear labels explaining each step and condition.

5. Question: Implicit and Explicit Type Conversion for Employee Bonus Calculation

Write a program that calculates an employee’s bonus based on their baseSalary (int) and performanceScore (double). Follow these steps:

- Convert baseSalary to double and calculate the bonus as baseSalary * performanceScore.
- Display the bonus as a double, and then cast it to an integer, rounding down. Print the integer bonus amount.
- If the integer bonus is greater than or equal to 1000, print "High Bonus"; if between 500 and 999, print "Moderate Bonus"; otherwise, print "Low Bonus."
- Also, use logical operators to check if performanceScore is over 1.5 or baseSalary is over 50000. If true, print "Exemplary Employee"; otherwise, "Standard Employee."

Each output should be accompanied by descriptive labels for clarity.