

Contents

Warm-up Exercises	1
Questions	5
Problems	9

Warm-up Exercises

1. Write a program that performs the following steps:
2. First ask user to enter any number between 0 and 10
3. Depending on the user's answer, do the following:

When user enters number 7, display the following message:

That's a lucky seven!

When user enters a number that is 5 or above but not 7, display the following message:

The number you entered **is** 5 or more.

When user enters a number that is below 5, display the following message:

The number you entered **is** less than 5.

Solution

```
Console.Write("Please enter a number between 0 and 100:
↵ ");
double input = double.Parse(Console.ReadLine());

if (input == 7)
{
    Console.WriteLine("That's a lucky seven!");
}
else if (input >= 5)
{
    Console.WriteLine("The number you entered is 5 or
↵ more.");
}
else
{
    Console.WriteLine("The number you entered is less
↵ than 5.");
}
```

1. Explain the meaning of the switch mechanism including the meaning of break and default. Give a small example.

Solution

A switch statement will execute code attached to a specific case determined by the value of a variable. The default keyword is used when defining the case in which the value of the determining variable does not match the value attached to any of the other cases.

Example

```
int i = 0
switch(i)
{
case 1: break;
case 2: break;
default: break;
```

1. Identify lines and cause of syntax errors preventing this code from execution #recursion

```
1: using System;
2:
3: namespace Factorial
4: {
5:     class Program
6:     {
7:         static int Factorial(int n); // returns the
↪ factorial value, or
8:         { // -1 if this value is not defined
9:             int answer = -1;
10:            if (n = 1) answer = 1;
11:            else if (n > 1) answer = n . Factorial(n - 1);
12:            return answer;
13:        }
14:        static void Main(string[] args)
15:        {
16:            int n = 7;
17:
↪ Console.WriteLine($"Factorial({n})={Factorial(n)}");
18:        }
19:    }
20: }
```

Solution

In line 7, there is a semicolon in between the method header and body.
In line 11, there is period in assignment statement.

1. Assume you are testing two positive (>0) integer numbers X and Y. What property is tested by during call of method E(X, Y) in the code given below?

```

using System;
class Program
{
    static bool E(int X, int Y)
    {
        if (((X % Y) == 0) && ((Y % X) == 0))
            return true;
        else
            return false;
    }
    static void Main(string[] args)
    {
        int X = 3; int Y = 2; // experiment with different
        ↪ values of X and Y
        {
            Console.WriteLine($"The answer for X={X} and
        ↪ Y={Y} is {E(X, Y)}");
        }
    }
}

```

Solution

The method E(X, Y) tests whether X and Y are equal.

1. Assuming a string variable name was declared and initialized with a value given by the user, write an **if** statement that displays "I have the same name!" if name matches your first name.

Solution

```

if (name == "John")
    Console.WriteLine("I have the same name!");

```

1. Write an **if-else** statement that displays "It's free for you!" if an **int** variable age is between 0 and 18, and "It's \$5.00." otherwise.

Solution

```

if (age > 0 && age < 18)
    Console.WriteLine("It's free for you!");
else
    Console.WriteLine("It's $5.00.")

```

1. Write a **switch** statement that sets a **double** variable discount to 0.5 if a **string** variable day contains "Monday" or "Wednesday", 0.25 if day contains "Saturday", and 0.125 otherwise.

Solution

```
switch(day)
{
    case "Monday":
    case "Wednesday":
        discount = 0.5;
        break;
    case "Saturday":
        discount = 0.25;
        break;
    default:
        discount = 0.125;
}
```

1. (We'll use the 24-hour clock, sometimes called "military time".) Assuming that an **int** variable `hours` has been initialized, write part of a program that would display on the screen "Good morning" if `hours` is less than or equal to 12, and "Hello" otherwise.

Solution

```
if (hours <= 12)
    {Console.WriteLine("Good morning!");}
else
    {Console.WriteLine("Hello");}
```

1. Assuming that `myString` is a **string** variable, write a statement that print "Hello, Mélodie!" if the value of `myString` is equal to "Mélodie", and nothing otherwise.

Solution

```
if (myString == "Mélodie")
    {Console.WriteLine("Hello, Mélodie!");}
```

1. Write a program that asks the user to enter a value between 0 and 10, and asks again as long as the user enters integers outside that range.

Solution

```
int answer;
do
{
    Console.WriteLine("Enter a value between 0 and 10
    ↵ (both included).");
    answer = int.Parse(Console.ReadLine());
}while(answer > 10 || answer < 0);
```

1. Write a small program that asks the user for an integer, and displays "It is positive" if the number entered is positive, "It is negative" if the number entered is negative, and "Not a number" if the user entered a string that is not an integer.

Solution

```
int answer;
Console.WriteLine("Enter an integer");
if(!int.TryParse(Console.ReadLine(), out answer))
{
    Console.WriteLine("Not a number");
}
else if (answer > 0)
{
    Console.WriteLine("Positive");
}
else
{
    Console.WriteLine("Negative");
}
```

1. Write a program that asks the user to enter a value between 1900 and 1999 (both included), and asks again as long as the user enters integers outside that range.

Solution

```
int answer;
do
{
    Console.WriteLine("Enter a value between 1900 and
↵ 1999 (both included).");
    answer = int.Parse(Console.ReadLine());
}while(answer < 1900 || answer > 1999);
```

Questions

1. Assume that A, B, and C are integer variables. What property on A, B, and C is tested by this code? Provide short justification.

```
if ((A-- == B++) && (--B == C++))
{
    Console.WriteLine($"Property satisfied");
}
else
{
```

```
        Console.WriteLine($"Property not satisfied");  
    }
```

Solution

This code tests whether or not the 3 variables are equal in value. Integers A and B are checked for equality before their values are incremented and decremented respectively. Then, B is incremented back to its original value and subsequently checked for equality with C.

1. The if-statement allows a program to run a piece of code based on the truth of some condition.

- Yes
- No

Solution

This code tests whether or not the 3 variables are equal in value. Integers A and B are checked for equality before their values are incremented and decremented respectively. Then, B is incremented back to its original value and subsequently checked for equality with C.

1. A(n) _____ statement allows a program to make a decision based on the truth or falsity of some condition.

- if
- test
- logic
- None of the above.

1. The three types of selection structures are:

- foreach, for and switch
- if, for and switch
- if, if-else and while
- if, if-else and switch.

1. In a switch statement, a case can be labeled as _____ so that it executes in the event that none of the provided cases are equivalent to the controlling expression.

- general
- default
- case *
- None of the above.

1. What is sequential processing?

Solution

When code is executed sequentially, without any branching.

1. What is a decision structure?

Solution

When portions of code are executed conditionally.

1. Determine if the following boolean expressions evaluate to **true** or **false**:
2. **true** || 3 > 4
3. 'A' == 'C' || ! **false**
4. (4 != 3) && **true**
5. (**true** && 4 >= 3) == **false**

Solution

1. true
2. true
3. true
4. false

1. What is the relational operator used to determine whenever two values are different?

Solution

!=

1. What is a flag?

Solution

A boolean variable.

1. Give three relational operators, and then two logical operators.

Solution

<=, ==, >, and !, &&

1. What would be displayed on the screen by the following code?

```
if ( false )  
{  
    Console.WriteLine("Hello!");  
}  
Console.WriteLine("Hi!");
```

Solution

"Hi!"

1. Is there a simpler way to write the expression `over21 == true`, assuming that `over21` is a boolean variable?

Solution

```
if(over21)
```

1. Assume that x and y are two **int** variables that have already been initialized (i.e. declared and assigned), write an **if** statement that assigns 10 to x if y is (strictly) greater than 5.

Solution

```
if ( y > 5 ) {x = 10;}
```

In C#, is there a difference between = and ==? Write a statement that uses =.

Solution

Yes, one is to assign, the other is to compare. **if** (x == 9){x = 12;}

1. Is the following statement correct, i.e., would it compile, assuming myFlag is a **bool** variable, and myAge is an initialized **int** variable?

```
if ( myAge > 20 )  
{  
    myFlag = true;  
};
```

Solution

No, the semi-colon should come before the }.

1. Write an **if** statement that prints "Bonjour!" if the value of the **char** variable lang is 'f'.

Solution

```
if ( lang == 'f' ) { Console.WriteLine("Bonjour !"); }
```

1. For each of the following boolean expressions, determine if it will evaluate to **true** or **false** when the boolean variables x, y, and z are all set to true:
 1. x || y && z
 2. !x || y && z
 3. !(x || y) && (z && y)
 4. (!x && x) || (x! || x) Do the same when they are all set to **false**.

Solution

1. true, false
 2. true, true
 3. false, false
 4. false, false
1. Write a boolean expression that evaluates to **true** if a variable x is between 3 (excluded) and 5 (included).

Solution

```
x>3 && 5>=x
```

1. Write an **if-else** statement that assigns 0.1 to z if y is greater than or equal to 0, and that assigns -0.1 to z otherwise.

Solution

```
if (y >= 0){z = 0.1;} else {z = -0.1;}
```

1. What will be displayed on the screen by the following program?

```
int x = 3, y = 2, z =4;
if (x > y) {z += y;}
if (x > z) {y -= 4;}
Console.WriteLine($"x is {x}, y is {y}, and z is {z}.");
```

Solution

"x is 3, y is 2, and z is 6."

1. What will be displayed on the screen by the following program?

```
int x = 3, y = 2, z =4;
if (x >= z) {z += y;} else if (x != y) {z *= y;}
y -= 4;
Console.WriteLine($"x is {x}, y is {y}, and z is {z}.");
```

Solution

"x is 3, y is -2, and z is 8."

1. What will be displayed on the screen by the following program?

```
int x = 3, y = 2, z = 4;
if (y >= z) {z += y;}
else if (x != y) { if (false) {z -= 3;} else {z += x;}}
Console.WriteLine($"x is {x}, y is {y}, and z is {z}.");
```

Solution

"x is 3, y is 2, and z is 7."

Problems

1. Write a program that decides whether you should sleep or not, based on whether you are tired and whether you have to work.

Solution

```
bool isTired = false;
bool dayOff = false;
```

```

    if (isTired && dayOff)
    {
        Console.WriteLine("You should get some sleep...");
    }
    else if (!dayOff)
    {
        Console.WriteLine("You don't need to sleep. Enjoy your
↪ day off!");
    }
    else if (!isTired)
    {
        Console.WriteLine("You have to work. No time for
↪ sleep.");
    }
    else
    {
        Console.WriteLine("You need to work today, and you've
↪ got the energy to do it. No need for sleep.");
    }

```

1. Rewrite, if possible, the three following **if-else-if** statements as **switch** statements:

```

// First if-else-if statement
if (myLang == 'f') { Console.WriteLine("Vous parlez
↪ Français ?"); }
else if (myLang == 'e') { Console.WriteLine("Do you speak
↪ English?"); }
else if (myLang == 'd') { Console.WriteLine("Sprechen Sie
↪ Deutsch?"); }
else { Console.WriteLine("I don't know your language!"); }

```

```

// Second if-else-if statement
if (myCity == "Augusta") { Console.WriteLine("I also live
↪ here!"); }
else if (myCity == "Paris" || myCity == "Boone")
{
    Console.WriteLine("I used to live there!");
}
else
{
    Console.WriteLine("I never lived there.")
}

```

```

// Third if-else-if statement
if (temp == 100.0) {Console.WriteLine("It's ready!"); }

```

```

else if (temp >= 90.0) {Console.WriteLine("Almost
↪ ready!"); }
else { Console.WriteLine("You have to wait."); }

```

If you think it is not possible or not feasible, explain why.

Solution for Part 1

```

switch (myLang)
{
    case 'f':
        Console.WriteLine("Vous parlez Français ?");
        break;
    case 'e':
        Console.WriteLine("Do you speak English?");
        break;
    case 'd':
        Console.WriteLine("Sprechen Sie Deutsch?");
        break;
    default:
        Console.WriteLine("I don't know your language!");
        break;
}

```

Solution for Part 2

```

switch (myCity)
{
    case "Augusta":
        Console.WriteLine("I also live here!");
        break;
    case "Paris":
    case "Boone":
        Console.WriteLine("I used to live there!");
        break;
    default:
        Console.WriteLine("I never lived there.");
}

```

Solution for Part 3

Impossible: can't write **switch** comparing all possible **float** values.

1. Write a program that asks the user to write a country name and stores the user's input into a string variable. Then, compare that string with **"france"**: if it is equal, then print "Bienvenue en France!". Then, compare that string with **"usa"**: if it is equal, then print "Welcome to the US!". If the string is different from both "france" and "usa", then print "Welcome to" followed by the name of the coun-

try the user typed in. Can you think of two ways to implement this program, one using **if-else-if** statements, the other using **switch**?

Solution for if-else-if Statement

```
Console.Write("Please enter the name of a country: ");
string country = Console.ReadLine();

if (country == "france")
{
    Console.WriteLine("Bienvenue en France!");
}
else if (country == "usa")
{
    Console.WriteLine("Welcome to the US!");
}
else
{
    Console.WriteLine($"Welcome to {country}!");
}
```

Solution for switch Statement

```
Console.Write("Please enter the name of a country: ");
string country = Console.ReadLine();

switch (country)
{
    case "france":
        Console.WriteLine("Bienvenue en France!");
        break;
    case "usa":
        Console.WriteLine("Welcome to the US!");
        break;
    default:
        Console.WriteLine($"Welcome to {country}");
        break;
}
```

1. You want to write a small program for an on-line printing company. Your program should ask the user to choose a format (10 x 15 centimeters, or 8 x 11 inches), ask if it is the first time the customer has ordered through your company, and the number of copies to print. Then, calculate the total cost of printing those pictures, knowing that
 - Printing a 10 x 15 -centimeter picture costs \$0.20, printing a 8 x 11-inch picture costs \$0.25
 - A new customer gets a \$3 coupon if the order is more than \$5

- A 10% discount is given if more than 50 copies were ordered
- The two previous offers can be cumulated. Display on the screen a message starting with "Welcome!", then a new line, then "We cherish our new customers" if it is the first time the user uses your company, ", so we're giving you a \$3 coupon!" if the user is allowed to get the coupon, then print the total and "You get a 10% discount!" if the user ordered more than 50 copies.

Solution

```

Console.WriteLine("Welcome! Is this your first time using
↪ our services? If so, type 'y'.");
string newUser = Console.ReadLine();

Console.WriteLine("Which format would you like to use?
↪ Enter 'c' for 10 x 15 centimeters, or anything else
↪ for 8 x 11 inches.");
string userFormat = Console.ReadLine();

decimal formatRate;
if (userFormat == "c") { formatRate = 0.2m; }
else { formatRate = 0.25m; }

Console.WriteLine("How many copies would you like?");
int copies = int.Parse(Console.ReadLine());
decimal total = formatRate * copies;

if (newUser == "y" && total > 5m)
{
    Console.WriteLine("Because we cherish our new
↪ customers, we're giving you a $3 coupon!");
    total -= 3m;
}

if (copies > 50)
{
    Console.WriteLine("You've earned a 10% discount for
↪ ordering more than 50 copies!");
    total *= 0.9m;
}

Console.WriteLine($"Grand Total: {total:C2}");

```

1. Write a program that asks the user to enter an integer between 0 and 100, and asks again until user enters one of these numbers (0, 1, ..., 99, or 100).

Solution

```

bool isInt;
int userInt;
do{
    Console.WriteLine("Please enter an integer between 0
↪ and 100.");
    isInt = int.TryParse(Console.ReadLine(), out userInt);
}while (!isInt || userInt < 0 || userInt >100)

```

1. You are writing a program for an ice-cream shop. At this ice cream shop customers have many options of what kind of ice cream they might get. The price of the ice cream depends on the options they choose.
- 1) First ask if they would like hard ice cream (\$1.50), soft-serve ice cream (\$1.00), or rolled ice cream (\$2.00).
- 2) If user chooses rolled ice-cream, it will always be served in a cup (\$0.50). Otherwise, ask user if they would like an ice cream in a cup (\$0.50) or a waffle cone (\$0.65).
- 3) Then ask user if they would like sprinkles. Only if they say yes add \$0.20 to price.
- 4) Lastly, display the final calculated ice cream price to the user.

Solution

```

decimal total = 0;
Console.WriteLine("Welcome to the ice cream shop!\n");
Console.WriteLine("What type of ice cream would you
↪ like?");
Console.WriteLine("1. Hard Ice-Cream - $1.50 \n2.
↪ Soft-Serve - $1.00 \n3. Rolled - $2.00\n");
Console.Write("Please select a number: ");
int userChoice = int.Parse(Console.ReadLine());

switch(userChoice)
{
    case 1:
        total += 1.5m;
        Console.WriteLine("\nWould you like that in a cup
↪ or waffle cone?");
        Console.WriteLine("1. Cup - $0.50 \n2. Waffle Cone
↪ - $0.65");
        Console.Write("Please select a number: ");
        userChoice = int.Parse(Console.ReadLine());
        break;
    case 2:
        total += 1m;

```

```

        Console.WriteLine("Would you like that in a cup or
↪ a waffle cone?");
        Console.WriteLine("1. Cup - $0.50 \n2. Waffle Cone
↪ - $0.65");
        Console.Write("Please select a number: ");
        userChoice = int.Parse(Console.ReadLine());
        break;
    default:
        total += 2m;
        Console.WriteLine("That will be served in a cup
↪ for $0.50.");
        userChoice = 1;
        break;
}

switch (userChoice)
{
    case 2:
        total += 0.65m;
        break;
    default:
        total += 0.5m;
        break;
}

Console.WriteLine("\nWould you like to add sprinkles for
↪ $0.20?");
string input = Console.ReadLine();
if (input.ToLower() == "yes")
{
    total += 0.2m;
}

Console.WriteLine($"Your total is: {total:C2}");

```

1. Write a program that:
2. Asks the user to enter a positive number between 0 and 100,
3. Asks again if user does not enter a number in this range,
4. Does not crash on bad user input, i.e. when input is not a number at all. (optional)

Example:

```

Enter a number between 0 and 100: No
Invalid input. Enter a number between 0 and 100: -20
Invalid input. Enter a number between 0 and 100: 5
You selected number 5.

```

Solution

```
double userInput;

do
{
    Console.WriteLine("Please enter a number between 0 and
↪ 100: ");
    userInput = double.Parse(Console.ReadLine());

    if (userInput < 0 || userInput > 100)
    {
        Console.WriteLine("Invalid input. ");
    };
} while (userInput < 0 || userInput > 100);

Console.WriteLine($"You selected the number {userInput}");
```

1. Write a program that will:
2. Ask the user to enter a positive number. Make sure user enters a number, and that that number is positive before you proceed with the next steps.
3. Find the largest number that is a multiplicand of 7, between 0 and the number user entered. Multiplicands of 7 are e.g. 7, 14, 21, 28, 35, 42, 49..... you want to find the one that is the closest, but not greater than the number user entered.
4. Display the result on the screen.

There is an edge case: If user enters a number that is less than seven, there is no answer. In that case the program should display appropriate output, such as "number you entered is less than 7".

Sample Output

```
Enter a positive number: -11
Try again. Enter a positive number: 52
The largest multiplicand of 7 is 49.
```

After you finish the program, test it with different numbers. For example:

- input 56 result is 56
- input 167 result is 161
- input 72364279 result is 72364278

Solution

```
double userNum;

do
{
```



```

    Console.WriteLine("Please enter a positive number: ");
    userNum = double.Parse(Console.ReadLine());

} while (userNum < 0);

int multiplicand = (int)userNum;
for (int i = 1; multiplicand % 7 != 0; i++)
{
    multiplicand -= i;
}

Console.WriteLine($"The largest multiplicand of 7 below
↪ {userNum} is {multiplicand}.");

```

1. Write a program that asks the user to enter a value between 0 and 10, and asks again as long as the user enters integers outside that range.

Solution

```

int userInt;
do
{
    Console.WriteLine("Please enter an integer value
↪ between 0 and 10");
    userInt = int.Parse(Console.ReadLine());
} while (userInt < 0 || userInt > 10);

```

1. Write a program that asks user to select an option A, B, or C. If user does not enter a valid selection, repeat the question until they do.

Solution

```

string userInput;
do
{
    Console.WriteLine("Please choose option A, B, or C.");
    userInput = Console.ReadLine();
} while (userInput != "A" && userInput != "B" &&
↪ userInput != "C");

```

1. Ask user to enter exactly 3 numbers (integers). Check that user enters valid numbers for each 3 inputs. Lastly display those numbers in order from largest to smallest. For example, if user enters numbers 16, 42 and 37, the program should output 42 37 16.

Solution

```

int userInt1, userInt2, userInt3;
bool valid = false;

```

```

do
{
    Console.WriteLine("Please enter the first integer: ");
    valid = int.TryParse(Console.ReadLine(), out
↪ userInt1);
}while (!valid);

do
{
    Console.WriteLine("Please enter the second integer: ");
    valid = int.TryParse(Console.ReadLine(), out
↪ userInt2);
} while (!valid);

do
{
    Console.WriteLine("Please enter the third integer: ");
    valid = int.TryParse(Console.ReadLine(), out
↪ userInt3);
} while (!valid);

if (userInt1 < userInt2)
{
    int temp = userInt2;
    userInt2 = userInt1;
    userInt1 = temp;
}
if (userInt3 > userInt2)
{
    int temp = userInt2;
    userInt2 = userInt3;
    userInt3 = temp;
}

Console.WriteLine($"{userInt1} {userInt2} {userInt3}");

```