# Independent Studies Tasks

***Simple Math, Data types, Operators***

Programming exercise I – Conversion

Write a program that prompts the user for a measurement in meters and then converts it into miles, feet and inches (displayed in form of “N meters = M miles” etc. via MessageBox). Use the following factors:

1 mile = 1610 meters

1 feet = 0.3048 meters

1 inch = 0.0254 meters

Programming exercise II – Circle

Write a program that prompts the user for a radius and then displays the area (S=π\*r^2) and circumference (L=2\* π\*r) of the circle with that radius (in form of “The area is N” via MessageBox)

Programming exercise III – Piggy bank

Write a program that contains a number of controls for user to enter the number of 1p coins, the number of 2p coins, the number of 5p coins, the number of 10p coins and the number of 20p coins contained in a piggy bank and calculates and displays the total amount of money, in pounds, contained in the piggy bank.

Programming exercise IV – Show digits

Write a program that accepts an integer (via textbox) and breaks it into a sequence of individual digits in reverse order. For example, the input 16384 is displayed as

4

8

3

6

1

You may assume that there are not more than five digits and input is not negative. Output should be given via MessageBox. (Hint: use Mod operator)

***Conditions***

Programming exercise V – Sorting

Write a program that prompts the user for three integers and then prints them in a sorted order (in a label). DO NOT use built-in sorting methods

Programming exercise VI – Advanced calculator

Your task is to write a program to calculate the sum of two numbers provided by the user in words (not by digits). You should put two textboxes to get the numbers and a button to add them up. The input can only be English one-digit numbers written in lower case (“zero”, “one”, “two”, “three”… up to “nine”). Your program should calculate the sum of those numbers and display the result via Label control.

Programming exercise VII – Tax code

Write a program that contains control so that user can enter a customers name, a purchase amount and a tax code. The tax code has been validated and will be one of the following:

* 1. tax exempt (0%)
  2. state sales tax only (3%)
  3. federal and state sales tax (5%)
  4. special sales tax (7%)

The program must then compute the sales tax and the total amount due and display information in a label in the following format

Customer name: NAME

Purchase amount: AMOUNT

Sales tax: N%

Total amount due: AMOUNT\_DUE

Programming exercise VIII – Grades

In the US education system letter grades (marks) are used to show the achievements of students. Letter grades A, B, C, D and F correspond to their numeric values 4, 3, 2, 1 and 0 accordingly. “A” is the highest letter grade a student may get.

Write a program that consumes user input (via TextBox), checks for validness (if the entered value is numeric and then if the input is between 0 and 4 inclusive) and then translates the number into the closest letter grade. For example, the number 2.8 would be converted to “B”.

Programming exercise IX – Cats

After many years of research, British scientists have found a formula to evaluate the attractiveness of cats. Attractiveness is measured in Purrs where 1 Purr is equal to attractiveness of a cat which is enough to make 1 human smile. Attractiveness depends on the current behaviour, appearance and age of a cat and can be evaluated as following:

where

A – attractiveness of a cat,

b – behaviour of a cat, equal to 5 if a cat is sleeping; 3.8 if a cat is playing; 1.2 if a cat is lying still,

h – length of hair, equal to 3.5 if at is long-haired and 1.3 if a cat is short-haired,

a – age of a cat, equal to 0.9 for kittens (under 6 months) and 0.1 for adult cats (6 months and more)

Your task is to create a program to evaluate attractiveness of a cat. You should use appropriate controls to grab user input for all variables (behaviour, age and length of hair) and calculate attractiveness using the formula above. User input should be validated. The output should be given via Label control in the following form:

This cat can make N people smile

Programming exercise X – Biorhythms

Biorhythms are used to predict various aspects of a person's life through simple mathematical cycles. Biorhythms start at the day of births and loop with the constant periods. There are three main biorhythms known:

* Physical, cycle is 23 days.
* Emotional, cycle is 28 days.
* Intellectual, cycle is 33 days.

It is well known that biorhythms can say how successful a relationship would be. Compatibility of a biorhythm can be calculated using reminder of number of days between birthdays divided over the length of a period. Let’s analyse the following example:

Husband’s birthday: 23/02/1987

Wife’s birthday: 31/05/1988

Since number of days between birthdays is equal to 463 (you can use Date.Substract() function and TimeSpan.TotalDays property to calculate the difference between two dates):

|  |  |  |  |
| --- | --- | --- | --- |
|  | Reminders | Mismatch  (reminder / length of period) | Compatibility  100% - Mismatch |
| Physical | 463 / 23 = 20, 3 in reminder | 3/23 = 13% | 100% - 13% = 87% |
| Emotional | 463 / 28 = 16, 15 in reminder | 15/28 = 54% | 100% - 54% = 46% |
| Intellectual | 463 / 33 = 14, 1 in reminder | 1/33 = 3% | 100% - 3% = 97% |

Your task is to create program that accepts two dates via DateTimePicker controls and shows compatibility of all three biorhythms.

Programming exercise XI – Temperature

Every day, a weather station receives 15 temperatures expressed in degrees Celsius. A program is to be written which will accept each Celsius temperature (via TextBox), convert it to Fahrenheit and display the converted temperature to the screen (via MessageBox). After 15 temperatures have been processed, the program should terminate.

Use the formula: °Fahrenheit = 1.8 \* °Celsius + 32 to convert degrees.

Programming exercise XII – Exchange rate

Write a program that assist currency exchange between US dollars and Uzbek sums.

The program should ask for exchange rate and US dollar value to convert (via two TextBoxes). Then the program should validate both inputs and convert USD value to sums (display using MessageBox). The program should terminate when the user enters 0 for exchange amount.

***Loops***

Programming exercise XIII – ATM

Write a program for the ATM (Automatic teller machine). The machine is asking user to enter its pin number (via TextBox) and then compares this number to the one in the database (assume that the number is “1234”).

If numbers match then welcoming message is displayed (you should display message “PIN is correct. Welcome!” via MessageBox), otherwise user has to re-enter the number.

If after the fifth attempt numbers are still mismatching, then the machine notifies user that the card is blocked (display message “Your card is blocked” via MessageBox).

You should validate user input:

* input should contain only numbers (hint: use *foreach* loop over the input and check if each character is a digit with char.IsDigit)
* input should contain exactly 4 numbers (both less and more should be considered as invalid input)

Invalid input is not counted as attempt.

Programming exercise XIV – Destiny number

You should calculate the “destiny number” – sum of all components (day, month, year) of a person’s date of birth. For example if you were born on 01/02/2003 your “destiny number” is 0+1+0+2+2+0+0+3 = 8

If the “destiny number” is more than 10 you should add up the digits of that number. If result is again more than 10 you should repeat the process until the result is between 1 and 9. For example for 31/08/1998 => 3+1+0+8+1+9+9+8 = 39 => 3+9 = 12 => 1+2=3

Programming exercise XV – Prime numbers

Your task is to figure out if the number entered by user is prime (prime number is a number that is only divisible over 1 and itself) or not.

***String manipulation***

Programming exercise XVI – Kaa language

You have formed a secret society with your mates and want to encrypt your communication so that no one can understand you talking. In order to do that you decided to alter all words to add “kaa” after each vowel (‘Y’ is not considered vowel), e.g.:

**Hi, how are you?**

becomes

**Hikaa, hokaaw akaarekaa yokaaukaa?**

Your task is to write a program that converts messages from and to Kaa language – user should be able to enter any text into a TextBox and click on Encode or Decode buttons to get the translation.

Programming exercise XVII – Change case

Your task is to write a program to adjust case of the text entered to be proper sentence case, e.g.

**hi, How are YOU? i’m fine, THankS.**

should be converted to

**Hi, how are you? I’m fine, thanks.**

Programming exercise XVIII – Time parsing

You task is to get number of seconds from the time string entered by user. For example:

1:45:30 (1 hour 45 minutes 30 seconds) = 1\*60\*60 + 45\*60 + 30 = 6330

2:55 (2 minutes 55 seconds) = 2\*60 + 55 = 175

As you can see user should not always provide all parts (hours, minutes and seconds). Input can only contain minutes and seconds, or just seconds.

You have to parse user input, calculate the total duration in seconds and show it via message box.

HINT: one of the possible solutions is to use TimeSpan class.

Programming exercise XIX – Prime filter

Prime number is a number that is only divisible over 1 and itself.

You are given (via TextBox control) a text containing some numbers separated by spaces e.g.

**1 3 5 6 10 13 17 18 20**

You program should process the input (HINT: use String.Split() function) to get the numbers, check which number is prime and show only prime numbers (in Label control) separated by space (HINT: consider using String.Join() function)

Programming exercise XX – Expression evaluation

Your task is to develop an expression processor, which should evaluate text containing simple arithmetic expressions. You should only support four operations – addition, subtraction, multiplication and division and only for one pair of numbers. So the task is to say what will be the sum of the two numbers entered by user e.g.:

User input: 25+35 => You should show: 60

User input: 99-7 => You should show: 92

User input: 12\*11 => You should show: 132

User input: 36/3 => You should show: 12

Programming exercise XXI – Sorting numbers

Your task is to write a program which accepts user input in a form of space-separated integers and sorts them in ascending order.

Input can only contain single-digit integers between 1 and 5 inclusive separated by space character. Any number can appear more than once (e.g. “1 5 5 2 3 1 4 2”). Any other characters are illegal; user should be notified by a friendly message in case of invalid input.

You should get the input via TextBox, validate it and give output in a Label control like following:

Input: 1 5 5 2 3 1 4 2

Output : 1 1 2 2 3 4 5 5

***Arrays***

Programming exercise XXII – Array sorting

Write a program that sorts an integer array and puts elements in the descending order.

The program should first ask for the number of elements in array and then ask user to feed the values for the array. After that the program should use the following algorithm for sorting:

1. Declare a Boolean variable hasChanges
2. Open a loop that should terminate then hasChanges contains false
3. Reset hasChanges variable to false
4. Go through all elements of the array and check if ith element is more than i+1th.
5. If this is the case – swap elements: declare separate variable, store the value of ith element in the variable, set ith element = i+1th, i+1th = the variable
6. If the was a swap change the value of hasChanges variable to true
7. Then sorting is done – display resulting array via Label