A Level Computer Science at Xaverian College

About the course:

We follow the AQA syllabus. The course is split into 3 modules:

Unit 1: Problem solving, Programming, Data Representation & Practical Exercise.

2 ½ hr On-Screen Exam - 40%

Unit 2: The Computer Components, Logic Gates, The Stored Program Concept and The Internet. **2 ½ hr written Exam -** 40%

Unit 3: NEA, coursework of your own choice – 20%

Stages of Problem Solving

One of the main aspects of Computing is problem solving. Solutions to problems are described and then automated by writing programs.

Stages in problem solving.

1. Understand the problem
2. Devise a plan
3. Carry out the plan
4. Review the solution

Example Problem:

Determine whether the numbers 231787 and 479561 are prime or not.

Understand the problem

What is a prime number?
A number that is only divisible by itself and 1.

Do we mean any number?

What do we mean by is divisible by?

How do we know that a number isn't prime?

Devise a plan

The plan

Assume the number is prime

If the number divides by 2 then number isn't prime

```
If the number doesn’t divide exactly
    Starting at 3
    Divide the number by successive odd numbers
    if the number divides by the divisor then the number isn’t prime
    repeat until number is not prime or divisor > square root of the number
```

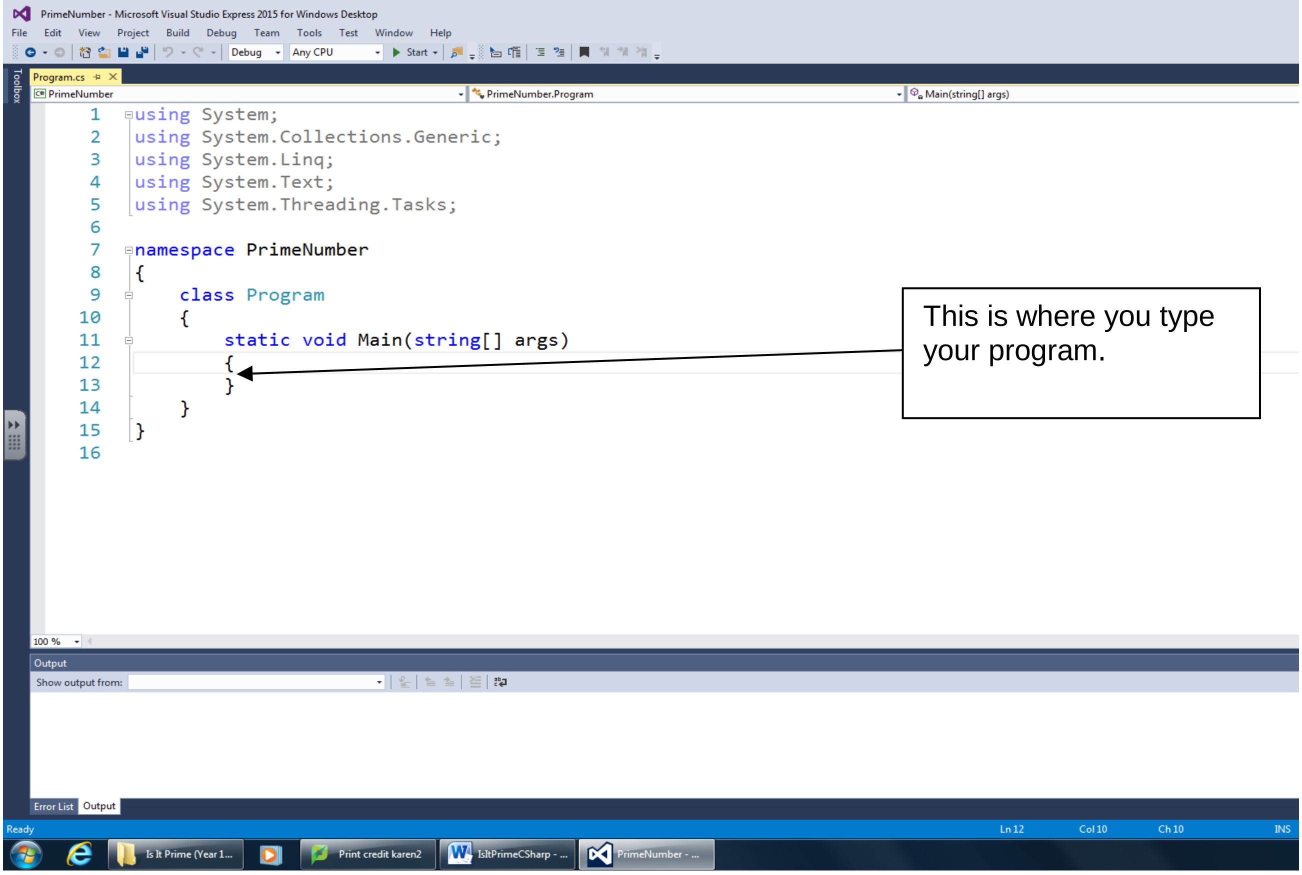
We now have a method for working out if the numbers are prime or not but it would take a long time for us to work through the plan. This is where automation comes in. What we want is a computer program to do the calculations for us.

The next stage would be to write a detailed algorithm – this is a step-by-step description of how to solve the problem. The program can then be developed from the algorithm.
The program is given later.

Carry out the plan.

We now need to type out the program and try it out.

Go to a computer. You should find the following screen is displayed.



```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace PrimeNumber
{
    class Program
    {
        static void Main(string[] args)
        {
        }
    }
}
```

This is where you type your program.

1) Between the **curly brackets** type out the following code:

```
Boolean IsPrime = true;
int Number = 231787;
if (Number % 2 == 0)
{
    IsPrime = false;
}
else
{
    int Divisor = 3;
    while (IsPrime == true && Divisor <= Math.Sqrt(Number))
    {
        if (Number % Divisor == 0)
        {
            IsPrime = false;
        }
        else
        {
            Divisor = Divisor + 2;
        }
    }
}
if (IsPrime == true)
{
    Console.WriteLine(Number.ToString()+" is prime");
}
else
{
    Console.WriteLine(Number.ToString()+" is not prime");
}
Console.Write("Press enter to exit.");
Console.ReadLine();
```

2

Note:

IsPrime, Number and Divisor are variables – memory locations in which the program will temporarily store values.

The program needs to know what sort of data is to be stored in the variable.

Number % 2 calculates the remainder when Number is divided by 2. If the remainder is 0 then Number is exactly divisible by 2 and so IsPrime is set equal to false.

Read the code carefully and see if you can work out what each command is doing.

3) Change the program so that the second number (479561) is tested. Run the program again and make a note of the output.

The program as it stands is quite limited. Every time we want to test a different number, we need to change the code. It would be better if we could enter the value to be tested while the program was running.

4) Replace the line

```
int Number = 479561;
```

with the following commands:

```
Console.Write("Enter number :");
int Number = Convert.ToInt32(Console.ReadLine());
```

Run the program again and test it with a number that is a prime and one that isn’t a prime.

The ReadLine command gets input from the keyboard. The input is a string (non-numeric) by default. It needs converting into an integer so that it can be stored in the variable Number.

Extensions:

The program only lets us test one number at a time. Every time we want to test a new number we have to re-run the program.

The program would be better if it allowed us to keep testing numbers until we chose to quit the program.

After the program has displayed the result of the test we want to instruct the user to enter 'q' if he/she wishes to quit the program.

5) Add a new declaration at the start of the program for the variable **`Reply`** as shown below:

```
string Reply = "";
```

6) Change the `Console.Write` at the bottom of the program so that it says "Enter q to quit.".

7) Change the `Console.ReadLine()` to `Reply = Console.ReadLine();`

We want the code to be repeated while Reply does not equal "q". To do this we need to enclose the code to be repeated between the lines:

```
do
  {
   //Code to be repeated
  }
 while (Reply != "q");
```

8) Add the lines given above so that the code to be repeated is enclosed. Run the program to see if it does as expected.

9) The program does not quite work correctly as it says that 2 and 3 are not prime. Try to see if you can correct this.