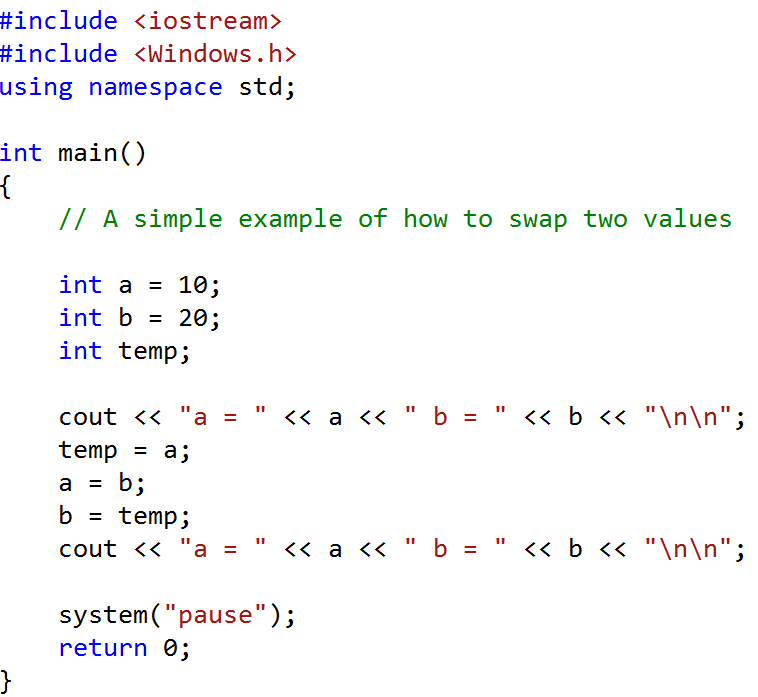
**Self Study Task Sheet**

**Task One**

Enter the following code and work through it to ensure you understand how it works.

****

**Task Two**

Write a program that accepts as input an integer value and then displays an appropriate message that says whether the number is odd or even.

**Task Three**

Write a program that accepts as input an integer value and then displays an appropriate message that says whether the number is positive, negative or zero.

**Task Four**

Write a program that accepts as input the user’s first name and surname and the prints out the appropriate message: your first name is longer than your surname; your surname is longer than your first name; your first name and surname are of equal length.

**Task Five**

Write a program that asks the user to enter the day of the week (Sunday, Monday, Tuesday, Wednesday, Thursday, Friday, Saturday or Sunday) and then displays an appropriate message. If the day is a Monday or Tuesday or Thursday or Friday the message should be ‘need to go to lectures today’ else if the day is a Wednesday the message should be ‘no lectures today I am going to revise’ else if the day is Saturday or Sunday the message should be ‘not sure if I am going to get up today.’ Your programme should display the message ‘never heard of that day before’ if the user mistakenly or intentionally types something other than a valid day.

**Task Six**

Write a program that requests the user enter their age and bank balance. If the bank balance is negative then the message ‘go for a walk’ should be displayed but if the bank balance is positive (or zero) and the user is 18 or over the message ‘go to pub’ should be displayed, if the bank balance is not negative and the user is under 18 the message should be ‘go to the cinema.’

**Task Seven**

Write a program that accepts as input three integer values and then using a sequence of if/then commands displays them in ascending order.

**Task Eight**

A company is hiring new recruits who should roughly be within the age range 18 to 28. Write a program that asks for the current year and year of birth of the user. The program should then display a message saying whether the user is eligible to apply or not.

**Task Nine**

The colours red, blue and yellow are known as primary colours because they cannot be made by mixing other colours. When you mix two primary colours you get a secondary colour, as shown below.

* When you mix red and blue, you get purple
* When you mix red and yellow, you get orange
* When you mix blue and yellow, you get green

Write a program that prompts a user to enter the names of two primary colours to mix. If something other than “red” “blue” or “yellow” is entered then an appropriate error message should be displayed else the secondary colour produced by the mixing should be displayed appropriately.

**Task Ten**

Write a program that asks the user to enter a year and then displays a message saying whether the year is a leap year on not. Use the following criteria to identify leap years.

* Determine whether the year is divisible by 100. If it is, then it is a leap year if and only if it is divisible by 400. For example, 2000 is a leap year but 2100 is not.
* If the year is not divisible by 100, then it is a leap year if and if only it is divisible by 4. For example, 2008 is a leap year but 2009 is not.

**Task Eleven**

A company sells a product that retails for £25 a unit. Quantity discounts are given according to the following table.

|  |  |
| --- | --- |
| **Quantity** | **Discount** |
| 10 – 19 | 20% |
| 20 – 49 | 30% |
| 50 – 99 | 40% |
| 100 or more | 50% |

Write a program that asks the user to enter the number of products bought and then displays the total cost before discount, the discount given, and the actual amount charged.

**Task Twelve**

Write a program that generates three random numbers within the range 0 through 100 and then displays them in ascending order.

**Task Thirteen**

Write and test a program that:-

* Prompts and reads a car’s engine size (in cubic cm) and if it is a Diesel or Petrol car.
* It then calculates the running costs as follows:
  + If the car is a Diesel the running costs for 100 miles are

£30 for cars with less than 1900 cm³   
£40 for cars with more than 1900 cm³ but 2100 cm³ or less

£50 for cars with more than 2100 cm³

* + If the car is a not a Diesel the running costs for 100 miles are

£35 for cars with less than 1900 cm³   
£45 for cars with more than 1900 cm³ but 2100 cm³ or less

£55 for cars with more than 2100 cm³

* Finally it displays the running costs

Example run of the program:

Input: Please enter the car’s engine size [in cubic cm]: 2000

Is it a Diesel [enter 1] or a petrol car [enter 2]: 1

Output: The running costs are £40

**Task Fourteen**

Write a program that displays the following menu and accepts an integer value as input:-

|  |
| --- |
| Geometry Calculator:   1. Calculate the area of a circle 2. Calculate the area of a rectangle 3. Calculate the area of a triangle 4. Quit     Enter your choice (1-4) : |

Depending on the users input the program should respond appropriately. For example, if the value ‘1’ was entered then the user should be prompted for the radius of the circle for which the area is required to be calculated, similarly if ‘2’ was entered the user should be prompted for the length and breadth of the rectangle for which the area is to be calculated. The program should then calculate the required area and display the result appropriately.

**Task Fifteen: Hamurabi game**

Continue the implementation of the Hamurabi game started last week by implementing the run for the first year (we haven’t done loops yet which are required for the following years) . Last week you coded the read-in of all the values that are required for running the game for the first year (e.g. how many bushels of grain do you want to use for planting, how much for feeding,…).

1. Using ‘if’ statements make sure that there is no impossible user input (e.g. negative numbers, more bushels of grain planted than you have, …)
2. Using calculations based on the game rules figure out how much grain is available at the end of the first year, and how many people are now living in your kingdom.
3. Using ‘if’ statements to decide if the game can continue or not (all people dead, no grain left)

We will develop this game further in the coming weeks.

**Task Sixteen: Grades**

Write a program that:

1. Prompts the user to enter a module percentage mark an then reads this percentage mark in
2. Figures out if the percentage mark equates to a fail (less than 40%), pass (between 40% and less than 60%), merit (between 60% and less than 70%) or distinction (70% or more).

**Task Seventeen: Las Vegas**

Write a program that:

1. Prompts the user to enter a number from the roulette table
2. Figures out if the number is red/black, high/low and odd/even

Modify your program so that the roulette number is generated randomly.

**Task Eighteen: Tic, Tac, Toe**

This is a tricky one. Write a program that:

1. Prompt two users to enter a ‘naught’ or a ‘cross’ respectively into one of the nine positions on the tic, tac, toe grid
2. After all 9 inputs have been made display the grid
3. To make it simpler: only when all 9 grid positions have been entered figure out if there is a winner (you can do this with a few ‘if’ statements)

We will develop this game further once we have introduced loops.

**Task Nineteen: FizzBuzz**

This is a classical task used in the recruitment for programming jobs. Apparently less than 1% of job candidates pass this test. Here is a slightly simplified version (the full version requires a loop that runs from 0 to 100, we do that in 2 weeks time):

1. Write a program that prompts the user to enter a number. Read it in.
2. For multiples of three print “Fizz”, for the multiples of five print “Buzz”. For numbers which are multiples of both three and five print “FizzBuzz”. For numbers that are neither just print the number.

Hint: consider using the modulus operator ‘%’.

End of Tutorial