**Object Oriented Programming**

**C# – Console**

**Your first object oriented program**

**Task 1**

1. Create a new class “Circle”
2. Create a public variable called **Radius** of type Integer
3. Create a Method called **Area** that returns the area of a Circle. Make use of Math.Pi to get Pi.
4. Create a Method called **Circumference** of type Double that returns the Circumference

**Task 2**

1. In the static void Main(string[] args) create a new instance of a Circle Object
2. Allow the user to set the Radius
3. Display the area of the Circle and the Circumference
4. Create a new Circle Object
5. Allow the user to set the Radius
6. Display the area of the Circle and the Circumference
7. Create a third Object as a reference Object that refers to one of the previous two circles
8. Alter the Radius of the reference Object and output the Area of each of the three circle Objects.

**Task 3**

1. Explore using separate classes and also using multiple classes in one Module (this is what the pre-release material is like)
2. Make notes on your observations.

The issues with all of the Object Oriented programs that we have created so far is that we have used Public access to variables. Our Variables should always be of Private nature so that other aspects of our programs can not alter the contents. Access to our variables must be set up in pre-defined ways. (Accessor Methods / Mutator Methods)

**Task 4**

1. Declare Radius as Private
2. Re-run the program – what errors do you get and why?
3. In the Circle Class create a new method that can be accessed from outside the class called: **SetRadius**. This should take a parameter r as Integer and assign its parameter to the Private Radius.
4. Alter the code in the static void Main to make use of the SetRadius Method.
5. Write a method to return the Radius and test this out from within the static void Main.

Constructors are special methods which allow control over the initialisation of objects. They are ran when an object is created (class is instantiated). In c# a constructor method is always stated as public **NameOfClass.** When using the New keyword we are also telling C# to allocate some memory for the class. This is really telling C# to construct the class.

Constructors are really useful for initialising variables and they run as soon as the object is created.

A constructor method looks like this:

public Circle()

{

Radius = 1;

}

public Circle(int r)

{

Radius = r;

}

**Task 5**

1. Create a constructor method for the Circle class and from within it give the radius an initial value of 1
2. Test a new circle object out with the initial value created during instantiation
3. Change the value in the constructor class and test the program again

Constructors can also be created with parameters and therefore be more powerful. We can set a value of for e.g. Radius during the object creation stage.

1. Add another constructor method that receives a parameter **r** as type **Integer**. The constructor method will set the Radius within the class to the value received in **r.**
2. Create (from void Main) a new instance of an Object Circle with a parameter value
3. Test the area and circumference

**Task 6**

Once you have checked your code for the above. Write a program which implements a class called Box that calculates the volume and surface area of a box given its height, width and length.

The volume is defined as the product of the height, width and length.

The surface area can be calculated by summing up the area of the six sides.

SurfaceArea = 2(lw+wh+lh)

Remember to create a class, start with variables and associated properties, followed by methods including any necessary constructors.

Use your Box class in Static Void Main() to show the range of methods and features implemented.

Print out your final code showing a range of Objects being used.