KAI JIE HI 100012607

Pass Task 4 – Referrence sheet

Related Learning Outcomes

ULO1 – Explain the OO Principles

This exercise demonstrated object encapsulation, learning the use of each operator and classes declaration

ULO2 – Use OO Language and Library

Demonstrated class and constructor declaration, the use of conditional statements (e.g. “if else, case,”), and assigning values to parameters, learn about operator or, and, not.

ULO3 – Design, Develop and Test using an IDE

The code was developed using Xamarin Studio to build and run the program, as well as integrated debugging features to step and inspect values.

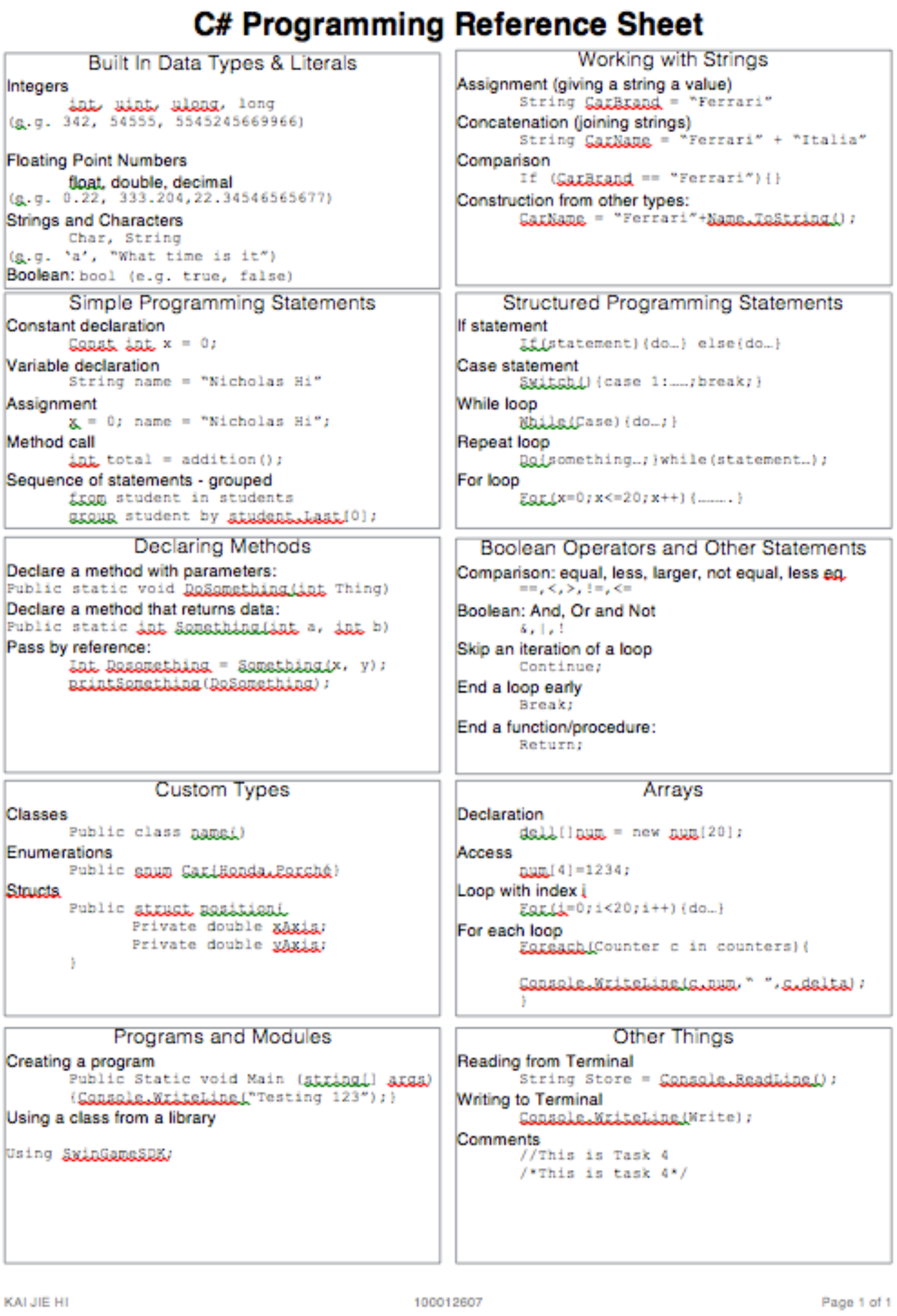
ULO4 – Communicate using UML Diagrams

I learned how to interpret a UML class diagram and write the related code.

ULO5 – Describe Elements of Good OO Design

The exercise demonstrated correct use of C# coding conventions.

Screenshots



C# Programming Reference Sheet

Built In Data Types & Literals

Integers

```
        int, uint, ulong, long
(g.g. 342, 54555, 5545245669966)
```

Floating Point Numbers

float, double, decimal

```
(g.g. 0.22, 333.204,22.34546565677)
```

Strings and Characters

```
        Char, String
(g.g. 'a', "What time is it")
```

Boolean: `bool (e.g. true, false)`

Working with Strings

Assignment (giving a string a value)

```
        String CarBrand = "Ferrari"
```

Concatenation (joining strings)

```
        String CarName = "Ferrari" + "Italia"
```

Comparison

```
        If (CarBrand == "Ferrari"){}
```

Construction from other types:

```
        CarName = "Ferrari"+Name.ToString();
```

Simple Programming Statements

Constant declaration

```
        Const int x = 0;
```

Variable declaration

```
        String name = "Nicholas Hi"
```

Assignment

```
        x = 0; name = "Nicholas Hi";
```

Method call

```
        int total = addition();
```

Sequence of statements - grouped

```
        from student in students
        group student by student.Last[0];
```

Structured Programming Statements

If statement

```
        If(statement){do…} else{do…}
```

Case statement

```
        Switch(){case 1:….;break;}
```

While loop

```
        While(Case){do…;}
```

Repeat loop

```
        Do{something…;}while(statement…);
```

For loop

```
        For(x=0;x<=20;x++){…….}
```

Declaring Methods

Declare a method with parameters:

```
Public static void DoSomething(int Thing)
```

Declare a method that returns data:

```
Public static int Something(int a, int b)
```

Pass by reference:

```
        Int Dosomething = Something(x, y);
        printSomething(DoSomething);
```

Boolean Operators and Other Statements

Comparison: equal, less, larger, not equal, less eq.

```
        ==,<,>,!=,<=
```

Boolean: And, Or and Not

```
        &,|,!
```

Skip an iteration of a loop

```
        Continue;
```

End a loop early

```
        Break;
```

End a function/procedure:

```
        Return;
```

Custom Types

Classes

```
        Public class name()
```

Enumerations

```
        Public enum Car{Honda,Porché}
```

Structs

```
        Public struct position{
                Private double xAxis;
                Private double yAxis;
        }
```

Arrays

Declaration

```
        dell[]num = new num[20];
```

Access

```
        num[4]=1234;
```

Loop with index i

```
        For(i=0;i<20;i++){do…}
```

For each loop

```
        Foreach(Counter c in counters){

        Console.WriteLine(c.num," ",c.delta);
        }
```

Programs and Modules

Creating a program

```
        Public Static void Main (string[] args)
        {Console.WriteLine("Testing 123");}
```

Using a class from a library

```
Using SwinGameSDK;
```

Other Things

Reading from Terminal

```
        String Store = Console.ReadLine();
```

Writing to Terminal

```
        Console.WriteLine(Write);
```

Comments

```
        //This is Task 4
        /*This is task 4*/
```