CST0006 – Computer Programming Foundations

INTRODUCTION to WEEK03

In this lecture and lab section we are going to go over all the things we learned in this weeks lecture. We will change spacing, use extra white space, go over semantics. We will use both Code::Blocks and Thonny for this lab, so if you haven't installed it yet, go through the previous lab in which we did this.

Before we begin, create a new Code::Blocks project and name it "2018W_CST0006_LAB04".

Run both "**Curly brace placement example 1**" and "**Curly brace placement example 2**" and answer question 1 on the answer sheet.

White Spaces

As we can see from the above example, for some languages, curly brace placement is purely cosmetic. Similarly, for some languages, white space and indentation is also purely cosmetic. Code Blocks allow you to group your ideas together so that you can visually separate them which will allow you to read your code solution in the way you have it formulated in your brain.

Here is an example of code that has removed the cosmetic white space.

```
#include <stdio.h>

int main(){int i=0;for(i=0;i<10;i++){printf("Number %d\n", i);}return 0;}
```

Run this code in Code::Blocks and answer question 2 on the answer sheet.

Indentation

Next we are going to run the Python example using "**Indentation example 1**" from the lecture notes in Thonny. For this part we are going to see what happens when you change the indentation.

Answer question 3 on the answer sheet.

White Spaces (Output)

As we've discovered, white space matters a lot. They allow us to organize code into clear thoughts and ideas. They are sometimes purely cosmetic and others a part of the syntactic structure of the language.

We also have to consider white spaces when we are expecting output too. For this part we are going to use the code example from "**Curly brace placement example 2**" in the lecture notes and play around with white space output in Code::Blocks.

Answer question 4 on the answer sheet.

Tabs vs Spaces

As we learned, not everyone uses the same coding conventions. We are going to assume we've been hired by a new company and the development team uses only 2 spaces for an indent and they use spaces instead of tabs.

We will now change these two options in Code::Blocks that will allow us to automatically use this new coding convention.

- Go to Settings in the menu, Choose Editor…
- Make sure "Use TAB character" is unchecked. (that forces tabs to become spaces)
- Change "TAB size in spaces:" to 2

After making these changes, go back to your code, and delete all the tabs then using the tab key, re-enter all the indents.

Answer question 5 on the answer sheet.

Semantics

As we discussed, sometimes the results of what you think you will get don't match what actually happened. Let's assume that we wanted a print out of Number 1 → Number 10, instead of Number 0 → Number 9, but we didn't code it properly.

So you now need to change the code so that you get this result.

Using "**Curly brace placement example 2**" change the code in Code::Blocks so that you get the required solution.

Copy and paste your solution to question 1 under Semantics on the answer sheet.

Using "**Indentation example 1**" change the code in Thonny so that you get the required solution.

Copy and paste your solution to question 2 under Semantics on the answer sheet.

Marking

1 mark for each group of questions answered

2.5 marks for each code semantics fixed