Practical-01: Linux Shell Commands, Version Control SVN and Basic Programming

Due 11 Aug by 22:00 **Points** 100 **Available** after 29 Jul at 0:00

Important!!!

Most of your practical activities have components with a wide range of interpretation. We use these components to assess your creativity to propose solutions, and capability to implementing these solutions. We highlight that are no wrong solutions. Moreover, we are highly interest in understanding your decisions and this process. Therefore, the explanation for your approach for solving these problems will have a high impact on your marks.

For more information about this, please see **Discussions/How does work Practical Marking? (click here)**

Introduction

This assignment will help to set the main concepts from this week. This assignment will focus on:

1. Linux shell command,
2. SVN and
3. Basic Programming.

Submission

Note: all class work must be submitted to your SVN repository. For this practice, you shall use:

```
https://version-control.adelaide.edu.au/svn/a<YOUR-ID>/2019/s2/fcs/week-01/practical-01/
```

Linux Shell Commands & SVN

Part 01 - Linux Commands

Problem 01

1. Using the concepts introduced on workshop-A, make the directory tree represented below;
2. List and perform standard input for every folder in the tree using '>' operator;
    - `ls > output.txt`, optionally you can also perform `ls > ls_current.txt` - whichever is easier for you.
3. Rename every subfolder from `sub_00X_folder_00N` to `subfolder_00N_00X`;
4. Move all the folders formated as `subfolder_00N_00X` into a single folder `<your_path_to>/practical-01/linux-command/part-01/folder_001/`;

Please, follow this directory tree:

```
+ practical-01
    + linux-commands
        + part-01
            - ls_current.txt
            + folder_001
                - ls_current.txt
                + sub_001_folder_001
                    - ls_current.txt
            + folder_002
                - ls_current.txt
                + sub_001_folder_002
                    - ls_current.txt
```

Part 02 - SVN

Problem 02

1. Use Linux command line to commit the folders you created in Part-01 to SVN. This task is very important to help you to acquired the agility and intuition of how SVN works.
    - Add folder by folder; This should increase the number of revision on your svn.
    - Perform a the commit;
    - Checkout in a different directory in your local computer. Show that you understand how working copies work locally.
        - Make sure you understanding the use of `svn status` to check whether the directory is a working copy.
    - Delete a few of the folders and commit this change; You can randomly select the folder to delete. The tutors will verify whether you perform this action by changing the versions on your SVN.
    - Add new files and commit your changes; You can randomly add files to your repository -- there is no right or wrong here.
    - Check if the WebSubmission system is capturing the new versions of your code; Make sure you are submitting multiple versions on your WebSubmission.

Part 03 - Basic Programming

Please note: ideally create a one folder per problem. Therefore, for problem-03 you can create.

```
+ problem-03/
    - <YOUR-FILE-NAME>.java
```

Although we are not assessing this file structure for the Java problems this Practical, it is very important that you get use to it for the remaining weeks of this course. Moreover, you are not required to commit any of your `*.class` files.

Problem 03

In Java, design and test the following task:

- Create a variable and storage the text "Hello world";
- Print on the screen the variable;

```
'''
save this project as problem-03/*.java
add to your svn repository
'''
```

Problem 04 - Summing and Printing

In Java, design and test the following task:

- Create variable number_a, number_b, integer type;
- Sum both numbers and attribute to variable C;
- Print on the screen "the sum of A and B is C" **Note:** A, B and C should be the value;

```
'''
save this project as problem-04/*.java
add to your svn repository

'''
```

Problem 05 - Multiplying and Printing

In Java, design and test the following task:

- Create variable number_a, number_b, integer type;
- Multiply both numbers and attribute to variable C;
- Print on the screen "The multiplication of A and B is C"; **Note:** A, B and C should be the value;

```
'''
save this project as problem-05/*.java
add to your svn repository
'''
```

Problem 06 - Division and Printing

In Java, design and test the following task:

- Create variable number_a, number_b, integer type;
- Divide both numbers and attribute to variable C, float type;
- Print on the screen "The division of A and B is C"; **Note:** A, B and C should be the value;

```
'''
save this project as problem-06/*.java
add to your svn repository
'''
```

Problem 07 - Subtraction and Printing

In Java, design and test the following task:

- Create variable number_a, number_b, integer type;
- Subtract B from A and attribute to variable C, float type;
- Print on the screen "The subtraction of A and B is C"; **Note:** A, B and C should be the value;

```
save this project as problem-07/*.java

add to your svn repository
```

Problem 08

In Java, consider the following information about a student:

- name, age, favorite food, grade (in maths and english), height, weight and number of subject

In this problem you are asked to define a software that ask for user input for all the aforementioned attributes (in the given order). It is your responsibility to define the data type of each of of those variables.

```
save this project as problem-08/*.java

add to your svn repository
```

Problem 09

In Java, consider the following information about a dog (poppy):

- name, age, favorite food, height, weight, favorite toy, and owner's name;

In this problem you are asked to define a software that ask for user input for all the aforementioned attributes (in the given order). It is your responsibility to define the data type of each of of those variables.

```
save this project as problem-09/*.java
add to your svn repository
```

Problem 10

Our university received an important message from an unknown secret agency from overseas. Although, our intelligence department from The University of Adelaide could partially crack the final message, it is still a mystery how we can organize this mysterious algorithm to run in a normal machine. In this problem, you are required to help the intelligence department to organize the code below. The University is counting on you...

```
        System.out.println(agent_message_01282);
        int agent_id = 007;
        System.out.println(agent_message_01282);
public class Main{
    public static void main(String [] args){
        agent_message_01282 = "My name is, Bond...";
        System.out.println(agent_message_01282);
}
        agent_message_01282 = "James Bond";
        String agent_message_01282 = "Who are you?";
        }
        agent_message_01282 = agent_message_01282 + "(00" + agent_id + ")";
        System.out.println(agent_message_01282);
```

Expected output:

```
Who are you?
My name is, Bond...
James Bond
James Bond(007)
```

```
save this project as problem-10/*.java
add to your svn repository
```

Practical 1

Criteria	Ratings				Pts
Part 1 Linux command	**20.0 Pts** **Excellent** You have created all files/directories correctly	**10.0 Pts** **Good** You have created most of files/directory correctly	**0.0 Pts** **No marks**		20.0 pts
Part 2 SVN	**20.0 Pts** **Excellent** You have committed all your works to SVN correctly	**10.0 Pts** **Good** You have committed most of your works to SVN correctly	**0.0 Pts** **No marks**		20.0 pts
Part 3 Basic programming	**60.0 Pts** **Excellent** 10 pts for each question. All of the following criteria are met: Codes are tidy and clean. Codes are bug-free. There are some comments to explain how it works.	**48.0 Pts** **Good** 8 pts for each question. At least one of the following criteria are met: Codes are tidy and clean. There are some comments to explain how it works. AND Codes are bug-free.	**30.0 Pts** **Fair** 5 pts for each question. The codes showing your understand but there is some bugs in it.	**0.0 Pts** **No marks**	60.0 pts
					Total points: 100.0