

Practical-Exam-01: Intro to Java, Conditions, and Loops

Due 18 Mar by 17:00 **Points** 100

Available 18 Mar at 15:00 - 18 Mar at 17:15 2 hours and 15 minutes

This assignment was locked 18 Mar at 17:15.

Introduction

This session ends 4:55 pm

Make sure to complete your submission in time.

Submission

- Create the repository on the SVN server
- You must checkout only the folder practical-exam-01 from your server
- **No other folders from your SVN will be allowed during the practical exam.**

```
cd

mkdir exams
cd exams

svn mkdir -m "first assessment commit" --parents https://version-control.adelaide.edu.au/svn/<YOUR-ID>/2022/s1/fcs/week-03/practical-exam-01

svn co https://version-control.adelaide.edu.au/svn/<YOUR-ID>/2022/s1/fcs/week-03/practical-exam-01

cd practical-exam-01
mkdir problem-01
mkdir problem-02
mkdir problem-03
mkdir problem-04
mkdir problem-05
svn add p*
```

Remember to svn add the Problem.java file inside each folder before you commit each solution

Assessment

- **This is a practical exam - your work must be entirely your own.**
- Marks for this practical exam contribute 2% to the final marks.
- Marks are for **evidence of participation**. Do at least try to commit answers to your repository and submit to web submission. Attempt as many questions as possible.

and submit to web submission. Attempt as many questions as possible.

- **Due date: 5 minutes from the end of this session.**
- **Do Not Forget** To Submit on [WebSubmission](https://myuni.adelaide.edu.au/courses/74996/modules/items/2617266)
(<https://myuni.adelaide.edu.au/courses/74996/modules/items/2617266>)
- **Late penalties:** Only work submitted during your enrolled practical session from a Linux/CS50 system in the practical lab will be marked.

Regarding functional marks, please consider:

- (1) only codes that can compile will be marked;
- (2) only codes that are in the suggested directory will be marked;
- (3) only codes submitted to SVN and WebSubmission before the deadline will be marked ;
- (4) only codes containing your signature on the top of the file will be marked by tutors;
- (5) you will have your mark decreased in 3 points if *.class file present in your folders;

Signature on your files

Note that all your coding files must contain on the top of it this information:

```
//=====
// Foundations of Computer Science
// Student: you name
// id: your id
// Semester:
// Year:
// Practical Exam Number:
//=====
```

Note

To acquire full marks

- (1) all your functionalities must be working perfectly,
- (2) your code has to be well and proportionally commented
- (3) your code must follow correct indentation (4 spaces for every new indentation level)
- (4) you have to use all the content from latest lectures.

We argue that you are not just asked to solve a problem but use the more sophisticated way to solve it. For instance, you can solve a problem with ten printing statements, but it will be better to solve the same problem with a printing loop.

Practical Exam 01

Part 01 - Basic Programming

Problem 01

In this problem you are required to design, implement and code a program that prints the following information on the command line console:

```
Today is 18th of March 2022 and this is my first practical exam.  
I do hope I can solve all the problems.
```

Constraints:

You will need to save your file in the problem-01 directory as below and add it using

svn add

Details:

```
file path: ../practical-exam-01/problem-01/Problem.java  
  
class name: Problem  
method to be implemented: public static void main(String [] args)
```

don't forget to commit your changes before accessing this assignment in websubmission.

Problem 02

In this problem you are required to design, implement and code a program that generates and prints on the screen the even integers between 10 to 30 (both inclusive).

```
[10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30]
```

Constraints:

```
file-path: ../practical-exam-01/problem-02/Problem.java  
  
class name: Problem  
method to be implemented: public static void main(String [] args)  
  
Note that, there are multiple ways to solving this problem. Please consider that  
for auto-prac-marker all the solutions will be awarded with the same weight. However,  
for tutor marking repetition structures will be awarded with total, and hard-coded  
solutions will be lightly weighted.
```

Problem 03

In this problem you are required to design, implement and code a program that asks the user for a length in inches. If the length typed by the user is in the range [0,400] you need to convert that

value to meters using the following relation:

1 inch = 0.0254 meters

The script must output the result with 3 decimal places (Hint: you need to use a printf statement).

If the length is not in that range, print a message saying the value is out of range as shown in the example below.

Expected output samples :

```
Enter a length in inches: 25
The length in meters is 0.635 m
```

Another run

```
Enter a length in inches: 500
The length 500 is out of range
```

Constraints

file-path: ..practical-exam-01/problem-03/Problem.java

class name: Problem

method to be implemented: public static void main(String [] args)

Note that, there are multiple ways to solving this problem. Please consider that for auto-prac-marker all the solutions will be awarded with the same weight. However, for tutor marking repetition structures will be awarded with total, and hard-coded solutions will be lightly weighted.

Problem 04

In this problem you are required to design, implement and code a program that repeatedly read two numbers (length and height) from the input and prints a rectangle of the given dimension.

When the user types a length of 0 (height is ignored then), the program stops

Sample output (blue text is the user input)

```
Enter width and height to draw a rectangular shape (type 0 to finish)
```

```
4 4
```

```
Drawing a square of size 4
```

```
####
```

```
####
```

```
####
```

```
####
```

```
Enter width and height to draw a rectangular shape (type 0 to finish)
```

```
7 2
```

```
Drawing a rectangle of size 7 by 2
```

```
#####
```

```
#####
```

Enter width and height to draw a rectangular shape (type 0 to finish)

0

Drawing is over

Constraints

file-path: ../practical-exam-01/problem-04/Problem.java

class name: Problem

method to be implemented: public static void main(String [] args)

script: make sure the prompt "Enter ..." finishes with a new line

Note that, there are multiple ways to solving this problem. Please consider that for auto-prac-marker all the solutions will be awarded with the same weight. However, for tutor marking repetition structures will be awarded with total, and hard-coded solutions will be lightly weighted.

Problem 05

In this problem, you are given several lines of a code that was already functional. However, this code was shuffled by some file transferring issue that occurred in our servers. You are asked to reorganize the code in order to make it functional once again, so that produces the expected output as shown below:

```
This is the last line of the code!
i: 1 j: 5 k: 5 - value 25
i: 2 j: 5 k: 5 - value 50
i: 3 j: 5 k: 5 - value 75
i: 4 j: 5 k: 5 - value 100
i: 5 j: 5 k: 5 - value 125
i: 5 j: 5 k: 6 - value 150
i: 5 j: 5 k: 7 - value 175
i: 5 j: 5 k: 8 - value 200
i: 5 j: 5 k: 9 - value 225
This is the first line of the code!
```

```
System.out.println("This is the first line of the code!");
System.out.println("This is the last line of the code!");
public static void main(String [] args){
    int ths = 25;
    for (int k = j; k < 10; k ++){
    }
    for(int i = 0; i < 10; i++){
        for(int j = i; j < 10; j++){
            int value = i * j * k;
        }
    }
    public class Problem{
        if(value % ths == 0 ){
            if(value != 0){
                System.out.print(" j: " + j);
                System.out.print("i: " + i);
                System.out.println(" k: " + k + " - value " + value);
            }
        }
    }
}
```

Constraints

file-path: ..practical-exam-01/problem-05/Problem.java

Note that, the following code was delivered to you not following any indentation. Consider that for tutor marking, we will assess whether you apply good Java conventions to the file.

Practical Exam 01 (1)

Criteria	Ratings				Pts
Functional	70 Pts Excellent Your code (1) perform all the functions correctly, (2) use latest concepts learned in class, (3) has a clear, creative and sophisticated way of solving the problem.	50 Pts Good Your code (1) perform all the functions correctly, (2) use concepts learned in class, (3) has a clear way of solving the problem.	40 Pts Fair Your code (1) perform almost all the functions correctly, (2) use concepts learned in class, (3) has a way of solving the problem.	0 Pts No marks You code (1) does not exist.	70 pts
Code Style	30 Pts Excellent Your code (1) has the right proportion of comments and place line and block comments correctly, (2) follow correct indentation every new indentation level, (3) has good variable naming, (4) has clear organization between tabs and is easy to read.	15 Pts Good Your code (1) has useful comments and place line and block comments correctly, (2) follow indentation, (3) has good variable naming.	5 Pts Fair Your code (1) has comments, (2) has variables, (4) has clear organization.	0 Pts No marks You code (1) does not exist.	30 pts
					Total points: 100