Download the source code for this lab, which is available on Canvas. Do not copy and paste code directly from this document. Your code may not compile!

Exercise One: Object references

1. After the following statements are executed, what are the values stored in each variable?

```java
int a = 7;
int b = 1;
int c = a + 2;
a = b;
b = c;
c = c + 1;
```

2. After the following statements are executed, what are the outputs? The method `setFruitName(String)` changes the fruit name and the method `getFruitName()` returns the fruit name of the object. You can find the `Fruit` class in the given source code for this lab.

```java
Fruit apple = new Fruit("red apple");
Fruit orange = new Fruit("orange");
Fruit greenApple = apple;

System.out.println("The fruit is " + apple.getFruitName());
System.out.println("The fruit is " + orange.getFruitName());
System.out.println("The fruit is " + greenApple.getFruitName());

orange.setFruitName("navel orange");
greenApple.setFruitName("green apple");

System.out.println("The fruit is " + apple.getFruitName());
System.out.println("The fruit is " + orange.getFruitName());
System.out.println("The fruit is " + greenApple.getFruitName());
```

Exercise Two: Method return types and parameters

Fill in the blanks below so that each method will compile.

1.

```java
private _____ getRandomLetter(String _____ ){
    int position = (int)(Math.random()*word.length());
    return word.charAt(position);
}
```

2.

```java
private _____ getSurname(_____ name) {
    int positionOfSpace = name.indexOf(" ");
    return name.substring(positionOfSpace + 1);
}
```

3.

```java
private _____ getBMI(double _____, ____ height){
    double bmi = weight / Math.pow(height,2);
    return bmi;
}
```

4.

```java
private _____ printTemperature(int _____ ){
    System.out.println("The temperature is " + degrees);
}
```

Exercise Three: Strings and Using String methods

1. What output do you think would be produced by each of the following code fragments?

   a. `System.out.println((int)2.9 * Double.parseDouble("4.5"));`
   b. `System.out.println("17" + Integer.parseInt("2") * 3.5);`
   c. `System.out.println("5 + 3" + 19 % 2 + 19 / 2);`
   d. `System.out.println(2 + 5 + "59" + 3 * 2 + (3 + 2));`

2. What is printed when the following start() method is executed?

```java
public void start() {

    String colours, first, second, third;
    int position1, position2, position3, length;

    colours = "redorangeyellow";

    first = colours.substring(4, 9);
    second = colours.substring(0, 4);
    third = colours.charAt(0) + colours.substring(13);

    length = third.length();
    third = third.toUpperCase();

    position1 = colours.indexOf('A');
    position2 = colours.indexOf("el");
    position3 = colours.indexOf("or");

    System.out.println("first: " + first);
    System.out.println("second: " + second);
    System.out.println("third: " + third);
    System.out.println("length: " + length);
    System.out.println("position1: " + position1);
    System.out.println("position2: " + position2);
    System.out.println("position3: " + position3);
}
```

Exercise Four: Debugging a simple program

The `CalculateVolume` program shown below is supposed to use the radius value entered by the user to calculate the volume of a sphere with the radius entered, then display the volume. The formula for the volume of a sphere is:

$$V = \tfrac{4}{3}\pi r^3$$

If, for example, a radius of 6.5 was entered at the prompt, the output should be:

```
"Volume of a Sphere"
Enter the radius: 6.5
Volume: 1150.3465099894624
```

There are **13** errors or omissions in the following source code. Locate and correct all of these so that the program would produce **exactly** the same output as above if a radius of 6.5 was entered. **Circle** the errors where they appear in the code below and write the corrected version:

```java
public class CalculateVolume

    public start() {

        double radius

        System.out.println("\"Volume of a Sphere"");

        System.out.println("Enter the radius: );

        radius = Integer.parseInt(Keyboard.readInput);

        int volume = 4 / 3 * Math.PI * Math.pow(radius,2);

        System.out.println("Volume: ", volume);

    }
```

Exercise Five: Determine lowest weight

Write a program which asks the user to enter the weights of two people, then uses `Math.min()` to determine the lowest weight, and prints out the result.

Note: You will need to use `Keyboard.readInput()` to read the values from the user and `Double.parseDouble()` to convert the values entered into floating-point numbers.

Here is an *example* of how your program should behave:

```
Enter first person's weight: 87.45
Enter second person's weight: 62.65
Lowest weight is 62.65.
```

The skeleton code is found in: `ictgradschool.industry.oop.lowestweight.LowestWeight.java`

Exercise Six: Generating and sorting 2 random numbers between 25 and 30

Write a program which generates 2 random integers between **25 and 30 (inclusive)**, then uses `Math.min()` and `Math.max()` to display them in **descending sequence**.

Here is how your output must be formatted. Note that the numbers displayed will no doubt differ from the *example* below as the numbers are randomly generated:

```
Your lucky numbers are 27 and 25
```

The skeleton code is found in: `ictgradschool.industry.oop.luckynumbers.LuckyNumbers.java`

Exercise Seven: Generating and sorting 3 random numbers

Write a program that prompts the user to enter a range – 2 integers representing a lower bound and an upper bound. Convert these bounds from `String` to `int` using `Integer.parseInt()`. Your program should then use `Math.random()` to generate 3 random integers that lie between the range entered (inclusive), and then use `Math.min()` to determine which of the random integers is the smallest.

Here is an example of how your program should behave:

```
Lower bound? 19
Upper bound? 42
3 randomly generated numbers: 29, 32 and 23
Smallest number is 23
```

The skeleton code is found in: `ictgradschool.industry.oop.sortingnumbers.SortingNumbers.java`

Exercise Eight: Removing a character from a sentence

Write a program that prompts the user to enter a sentence, then prints out the sentence with a random character missing. The program is to be written so that each task is in a separate method.

Here is an example of how your program should behave:

```
Enter a sentence: Have a nice day!
Removing "v" from position 2
New sentence is: Hae a nice day!
```

You need to write 5 methods, one method for each of the following tasks:
- Printing the prompt and reading the input from the user: *getSentenceFromUser()*
- Generating an appropriate random number: *getRandomPosition(String sentence)*
- Printing the character that is going to be removed: *printCharacterToBeRemoved(String sentence, int randomPosition)*

- Generating a new sentence by removing the character at the random position: *removeCharacter(String sentence, int randomPosition)*
- Printing the new sentence: *printNewSentence(String changedSentence)*

Here is an example of using these methods so that the program behaves according to the previous example:

```java
String sentence = getSentenceFromUser();
int randomPosition = getRandomPosition(sentence);
printCharacterToBeRemoved(sentence, randomPosition);
String changedSentence = removeCharacter(sentence, randomPosition);
printNewSentence(changedSentence);
```

The skeleton code is found in:
`ictgradschool.industry.oop.removechar.RemoveCharacter.java`

Exercise Nine: Truncating an amount to n decimal places

Write a program that prompts the user to enter an amount and a number of decimal places. The program should then truncate the amount to the user-specified number of decimal places using `String` methods.

Here is an example of how your program should behave:

```
Please enter an amount: 45.893
Please enter the number of decimal places: 1
Amount truncated to 1 decimal places is: 45.8
```

To truncate the amount to the user-specified number of decimal places, the `String` method `indexOf()` should be used to find the position of the decimal point, and the method `substring()` should then be used to extract the amount to the user-specified number of decimal places. The program is to be written so that each task is in a separate method. You need to write four methods, one method for each of the following tasks:
- Printing the prompt and reading the amount from the user
- Printing the prompt and reading the number of decimal places from the user
- Truncating the amount to the user-specified number of decimal places
- Printing the truncated amount

The skeleton code is found in:
`ictgradschool.industry.oop.truncateamount.TruncateAmount.java`

Exercise Ten: Noughts and crosses

Write a program to extract 3 rows of 6 characters from a `String` of 18 characters. The program should print out the 3 rows, followed by the left diagonal of the 3 rows. The `String` method `substring()` can be used to extract the required characters.

The program is to be written so that each task is in a separate method. You need to write 4 methods, one method for each of the following tasks:

- Extracting and returning a specified substring of 6 characters from the `String`
- Printing the 3 rows
- Extracting and returning the left diagonal
- Printing the diagonal

Here is an example of the output of the program:

```
X X O
O X O
X O X
Diagonal: X X X
```

The methods will be used as follows for the program:

```
String letters = "X X O O X O X O X ";

String row1 = getRow(letters, 1);

String row2 = getRow(letters, 2);

String row3 = getRow(letters, 3);

printRows(row1, row2, row3);

String leftDiagonal = getLeftDiagonal(row1, row2, row3);

printDiagonal(leftDiagonal);
```

The skeleton code is found in:
`ictgradschool.industry.oop.noughtsandcrosses.NoughtsAndCrosses.java`

(Challenge) Exercise: Scrambling the letters in a word

There is no skeleton code for this exercise.

Write a program that prompts the user to enter a word. The program should then randomly rearrange all the letters in the word, then ask the user to choose one of the letters in the word. The new position of this letter will be displayed, and the scrambled word displayed on the next line.

Here is an *example* of how your program should behave:

```
Enter a word: caterpillar
Choose a letter: p
This letter is now in position 3
The scrambled word is arepcrialtl
```

The program is to be written so that each task is in a separate method. You need to write 5 methods, one method for each of the following tasks:
- Printing the prompt, reading the word from the user, then returning the word
- Randomly rearranging all the letters in the word and returning the scrambled word
  You may use the following method for rearranging the letters

```java
private String rearrangeWord(String word) {
    String lettersRemaining = word;
    String newWord = "";
    for (int i = 0; i< word.length(); i++) {
        int randomPosition = (int)(Math.random() *
                            lettersRemaining.length());
        newWord += lettersRemaining.charAt(randomPosition);
        lettersRemaining =
        lettersRemaining.substring(0, randomPosition) +
            lettersRemaining.substring(randomPosition + 1);
    }
    return newWord;
}
```

- Printing the prompt, obtaining the chosen letter from the user, then returning the letter
- Printing the position of the chosen letter in the scrambled word
- Printing the scrambled word