CSE 215: Programming Language II Lab

Lab – 3 **Lab Officer: Tanzina Tazreen**

Methods

Objective:

- To learn about methods
- To learn to implement a program using multiple methods

A **Method** is a block of code which only runs when it is called.

- You can pass data, known as parameters, into a method.
- A method may or may not "return" data after processing.
- Methods are used to perform certain actions, and they are also known as **functions**.
- A method must be declared within a class.

Why use methods?

To reuse code: define the code once, and use it many times.

Method structure

```
<:static> <return_type> <method_identifier> (params)
{
    // statements
}
```

`:static` -> means that the method belongs to the Main class and not an object of the Main class.

`return_type` -> The type of data the method returns. It may be any of the primitive data types, arrays or may not return anything at all (void).

`method_identifier` -> Name of the method.

`params:` Input arguments to the method. The number o arguments can be 0 to many.

```
public class Main {
  //method without any parameter
  static void myMethod() {
    System.out.println("Method with no param just got
executed!");
  }

//Method with one parameter
```

```
static void myMethod(String fname) {
    System.out.println(fname + " is coding");
  }

//Method with multiple parameter
static void myMethod(String fname, int age) {
    System.out.println(fname + " is " + age);
  }

  public static void main(String[] args) {
    myMethod();
    myMethod("Liam");
    myMethod("Liam", 5);
  }
}
```

```
Method with no param just got executed!
Liam is coding
Liam is 5
```

Method Overloading

method overloading is when multiple methods have the same name with different parameters. Multiple methods can have the same name as long as the number and/or type of parameters are different (return type does not matter).

Instead of defining two methods that should do the same thing, it is better to overload one.

```
public class Main {
  static int add(int x, int y) {
    return x + y;
  }

  static double add(double x, double y) {
    return x + y;
  }

  public static void main(String[] args) {
    System.out.println(add(8, 5));
    System.out.println(add(4.3, 6.26));
  }
}
```

```
13
10.559999999999999
```

Task:

1. Write a method `countOdd(int[] array1, int[] array2)` that takes two integer array as parameters and returns one of the two array which has the most number of odd numbers as its elements. In the main method call that method and print only the elements which are odd Numbers.

3. Write a program that has the following static variable

`balance (initial value 0)`
and these static methods:
`deposit(double amount):` Increase account balance
`withdraw(double amount):` Decrease account balance.
User cannot withdraw if `amount > balance`, so display an appropriate message in this particular case if it happens.

1. Deposit
2. Withdraw
3. Balance
4. Exit

Now run an infinite loop in main program so it displays user with following options:
Under options 1 and 2, the program should ask for appropriate user input (i.e. amount to deposit).
Display `balance` variable if user chooses option 3.