



**MACQUARIE**  
University

*Department of Computing*

**ITEC625 Fundamentals of Computer Science**  
**Workshop - Methods**

## Learning outcomes

This weeks workshop is about understanding defining and calling methods.

### 1. Identify need for a method

Consider each of the following codes and identify if there is any need for a method?

(a)

```
int a = 7, b = 9, c;
if(a > b) {
    c = 1;
}
else if(a < b) {
    c = -1;
}
else {
    c = 0;
}

int p = 5, q = 4, r;
if(p > q) {
    r = 1;
}
else if(p < q) {
    r = -1;
}
else {
    r = 0;
}

int x = 8, y = 8, z;
if(x > y) {
    z = 1;
}
else if(x < y) {
    z = -1;
}
else {
    z = 0;
}
```

```
public class Client {  
    public static int compare(int n1, int n2) {  
        if(n1 > n2) {  
            return 1;  
        }  
        if(n1 < n2){  
            return -1;  
        }  
        return 0;  
    } //end compare  
  
    public static void main(String[] args) {  
        int a = 7 , b = 9;  
        int c = compare(a, b);  
  
        int p = 5, q = 4;  
        int r = compare(p, q);  
  
        int x = 8, y = 8;  
        int z = compare(x, y);  
    } //end main  
} //end class
```

(b)

```
int a = 57, b = 10;
int remainder = a%b;
int n = a - remainder;
if(remainder >= b/2) {
    n = n + b;
}

int p = 17, q = 5;
remainder = p%q;
int r = p - remainder;
if(remainder >= q/2) {
    r = r + q;
}
```

```
public class Client {
    public static int roundOff(int n1, int n2) {
        remainder = n1%n2;
        int result = n1 - remainder;
        if(remainder >= n2/2) {
            result = result + n2;
        }
        return result;
    } //end roundOff

    public static void main(String[] args) {
        int a = 57, b = 10;
        int n = roundOff(a, b);

        int p = 17, q = 5;
        int r = roundOff(p, q);
    } //end main
} //end class
```

## 2. Designing functions Based on the lecture notes, draw a block diagram for methods,

- when passed a number, returns `true` if the integer is even, and `false` otherwise. What is the data type for the input(s)? What is the data type for the return value/ output?  
solid arrow with text label "int" going from the caller to the method block "isEven" and dashed arrow with text label "boolean" going back to the caller.
- when passed two numbers, returns `true` if they both have the same last digit, and `false` otherwise. What is the data type for the input(s)? What is the data type for the return value/ output?  
solid arrow with text label "int, int" going from the caller to the method block "sameLastDigit" and dashed arrow with text label "boolean" going back to the caller.
- when passed three boolean values, return the number of values that are `true`. What is the data type for the input(s)? What is the data type for the return value/ output?  
solid arrow with text label "boolean, boolean, boolean" going from the caller to the method block "countTrues" and dashed arrow with text label "int" going back to the caller.

3. **Calling methods** Consider each of the following methods and write down a method call with actual parameters of your choice. You may assume that the calls are made from another method in the same class.

Make sure you store the value returned (if any) in a variable of the correct data type and state the value of that variable as a comment.

The first part is solved as an example.

(a)

```
public static int nDigits(int n) {  
    int result = 0;  
    while(n!=0) {  
        result++;  
        n=n/10;  
    }  
    return result;  
}
```

**SOLUTION** - Method call:

```
int a = 452017;  
int k = nDigits(a); //k will be 6
```

(b)

```
public static boolean and(boolean a, boolean b) {  
    if(a == false)  
        return false;  
    if(b == false)  
        return false;  
    return true;  
}
```

```
boolean a = true, b = false;  
boolean c = and(a, b); //c is false
```

(c)

```
public static int roundOff(double a) {  
    int b = (int)a;  
    double decimal = a - b;  
    if(decimal < 0.5)  
        return b;  
    else  
        return b+1;  
}
```

```
double a = 31.6;  
int b = roundOff(a); //b is 32
```

(d)

```
public static double average(int a, int b) {  
    return (a+b)/2.0; //not 2, but 2.0  
}
```

```
int a = 4, b = 7;  
double c = average(a, b); //c is 5.5
```

#### 4. Defining a method from scratch

Write down the definition for each of the methods for which you drew a block diagram in question 2.

```
public static boolean isEven(int a) {
    if(a%2 == 0) {
        return true;
    }
    else {
        return false;
    }
}

public static boolean sameLastDigit(int a, int b) {
    if(a%10 == b%10) {
        return true;
    }
    else {
        return false;
    }
}

public static int countTrues(boolean a, boolean b, boolean c) {
    int result = 0;
    if(a)
        result++;
    if(b)
        result++;
    if(c)
        result++;
    return result;
}
```

#### 5. Implementation given method headers.

Import the project from archive file `methodsWorkshopTemplate` and implement as many methods as you can. Tests are provided in in class `AllInOneTest`.

We haven't provided a header for the method `sameOddity`. Read the specification and write down the method header and body. Then uncomment the corresponding test in class `AllInOneTest`.

solutions provided in archive file `methodsWorkshopSolution`