

## Objects and Classes

### methods, constructors, access control

#### Objectives:

After completing the following exercises, students will be able to:

1. Declare methods with/without return value and with/without parameters
2. Write programs with calls to methods of predefined and user-defined classes
3. Trace programs with method calls

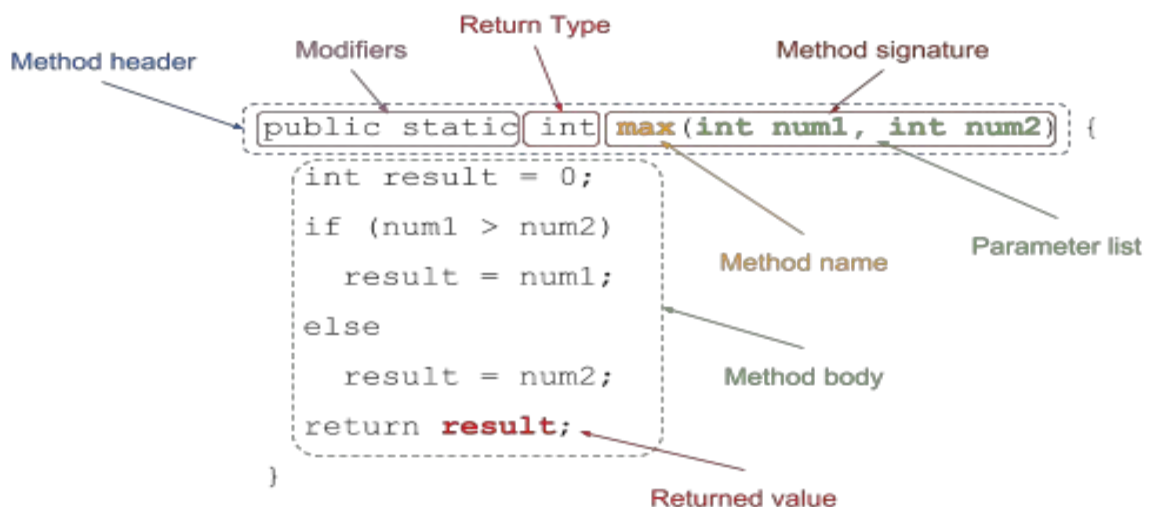
#### Exercise 1:

Given the following method, point out the following:

- a) Modifier
- b) Return type
- c) Returned value
- d) Method header
- e) Method signature
- f) Method name
- g) Parameters list
- h) Method body

```
public static int max(int num1, int num2) {  
    int result = 0;  
    if (num1 > num2)  
        result = num1;  
    else  
        result = num2;  
    return result;  
}
```

Answer:



## Exercise 2:

Show the output of the following program:

```
class Test {
    public static void method1(int i, int num) {
        for (int j=1; j <= 1; j++) {
            System.out.print(num + " ");
            num *= 2;
        }
        System.out.println();
    }

    public static void main(String[] args) {
        int i = 1;
        while (i <= 6) {
            method1(i, 2);
            i++;
        }
    }
}
```

Answer:

```
2
2 4
2 4 8
2 4 8 16
2 4 8 16 32
2 4 8 16 32 64
```

### Exercise 3:

Implement the class Time in Java

Attributes:

- **sec**: seconds between 0 and 59
- **min** : minutes between 0 and 59
- **hour** : can be any positive integer

Methods

- **readTime** : reads values of sec, min and hour from the keyboard
- **fixTime** : assures that sec and min are in the appropriate ranges
- **toSec** : convert the time to seconds
- **addSec** : increases the seconds by amount s, keeping sec and min within ranges
- **addMin** : increases the minutes by amount m keeping sec and min within ranges
- **addHour** : increases the hours by amount h
- **addTime** : increases the time by sec, min, hour of t
- **display** : print the attribute values in the format: hour:mm:ss

Time
+ sec : 0..59
+ min : 0..59
+ hour : integer
+ readTime()
+ fixTime()
+ toSec(): integer
+ display()
+ addSec(s : 0..59)
+ addMin(m : 0..59)
+ addHour(h : integer)
+ addTime(t : Time)

Answer:

```
import java.util.Scanner;

class Time {

    public int sec, min, hour;

    public void readTime() {
        Scanner S = new Scanner(System.in);
        System.out.print("Enter the seconds: ");
        sec = S.nextInt();
        System.out.print("Enter the minutes: ");
        min = S.nextInt();
        System.out.print("Enter the hours: ");
        hour = S.nextInt();
        fixTime();
    }

    public void fixTime() {
        min += sec / 60;
        sec %= 60;
        hour += min / 60;
        min %= 60;
    }

    public int toSec() {
        int result;
        result = sec + min*60 + hour*3600;
        return result;
    }

    public void addSec(int s) {
        sec += s;
        fixTime();
    }
}
```

```

public void addMin(int m) {
    min += m;
    fixTime();
}

public void addHour(int h) {
    hour += h;
}

public void display() {
    String result = hour+": ";
    if (min < 10) result += "0";
    result += min+": ";
    if (sec < 10) result += "0";
    result += sec;
    System.out.println(result);
}

public void addTime(Time t) {
    addSec(t.sec);
    addMin(t.min);
    addHour(t.hour);
}
} // end of class

```

#### Exercise 4:

Write a program that uses class Time to do the following:

- create two objects t1 and t2 of class Time and read their information
- increase t1 by 37 seconds and t2 by 15 minutes
- Tell whether t1 occurs after t2 or not
- increase t1 by t2 and print it

Answer:

```

class TestTime {
    public static void main(String[] args) {

        Time t1 = new Time();
        t1.readTime();

        Time t2 = new Time();
        t2.readTime();

        t1.addSec(37);
        t2.addMin(15);

        if (t1.toSec() > t2.toSec())
            System.out.println("t1 occurs after t2");
        else
            System.out.println("t1 does not occur after t2");

        t1.addTime(t2);
        t1.display();
    }
}

```