Functions

methods basic

- each prgram has one and only one main method
- in Java, method name can not start with 'Number', '#' or '-'. also can not be a reserved word

Exercises

1. which method can only be called once in a program?
   a) main method
   b) static method
   c) finalize method
   d) private method

---

2. in java programming, which of the following is not a valid method name?
   a) main
   b) 1main
   c) main1
   d) main()

Calling a function

- you can call a method one or more times

Exercises

3. which statment should we put in blank to print *hello world*.

```
//method declaration
public static void nameForMethod() {
    System.out.println("Hello World");
}
public static void main(String[] atgs) {
    //method calling
    ________________________;
}
```

```
  a) nameForMethod(1)
  b) nameForMethod(int i)
  c) nameForMethod()
  d) nameForMethod
```

---

4. fill the blanks to print three different sentences, announcing name of three great chef.

```
//method declaration
public static void printName(String name) {
    System.out.println(name + "is a great chef");
}
public static void main(String[] args) {
    //method calling
    ___________"Nusret"_____;
    ___________"Gordon"_____;
    ___________"Guy Fury"_____;
}
```

---

method parameters

- a method can have none, one or more parameters
- parameters are separated by commas

Exercises

5. which statment should we put in blank to check if number '5' is even

```
//method declaration
public static int isOdd(__________) {
    if (number % 2 == 0) {
        return 0;
    } else {
        return 1;
    }
}
public static void main(String[] args) {
    //method calling
    System.out.println(isOdd(5));
}
```

```
  a) int number
  b) int num
  c) number
  d) 5
```

---

6. what is the output of the following code

```
//method declaration
public static int add(number1, number2) {
    return number1 + number2;
}
public static void main(String[] args) {
    //method calling
    int sum = add(1, 2);
    System.out.println(sum);
}
```

---

7. what's wrong with the following code

```java
//method declaration
public static simpleCalculator(char oprator; int number1; int number2) {
    if (oprator == '+') {
        return number1 + number2;
    } else if (oprator == '-') {
        return number1 - number2;
    } else if (oprator == '*') {
        return number1 * number2;
    } else if (oprator == '/') {
        return number1 / number2;
    } else {
        return "invalid operator";
    }
}
```

Return values

- a method can return a value
- the return type of the method must match the return type of the value
- if the return type is void, then the method does not return a value

Exercises

8. what should be in the blank to return if the number is prime or not.

```java
//method declaration
public static boolean isPrime(int number) {
    for (int i = 2; i < number; i++) {
        if (number % i == 0) {
            return ______;
        }
    }
    return _____;
}
```

---

9. what's the problem with the following code

```java
//method declaration
public static void max(int number1, int number2) {
    if (number1 > number2) {
        return number1;
    } else {
        return number2;
    }
}
```

---

passing arguments

- when you pass an argument to a method, the argument is passed by value this means that the method receives a copy of the value of the argument therefore, any changes made to the parameter inside the method have no effect on the argument

  Exercises

  10. what's the output of the following code

```
//method declaration
public static void change(int number) {
    number = 10;
}
public static void main(String[] args) {
    int number = 5;
    change(number);
    System.out.println(number);
}
```

a) 5

b) 10

c) logic error

d) compile error

____

11. how can we change the code to make it work

```
//method declaration
public static void change(int number) {
    number = 10;
}
public static void main(String[] args) {
    int number = 5;
    _______________
    System.out.println(_______);
}
```

____

Overloading methods

- you can define multiple methods with the same name as long as the number and/or type of the parameters are different

  Exercises

  12. what's the output of the following code

```
//method declaration
public static void printName(String name) {
    System.out.print(name);
}
public static void printName(String firstName, String lastName) {
    System.out.print(firstName + " " + lastName);
}
public static void main(String[] args) {
    printName("Nusret ");
    printName("Nusret", "Gökçe");
}
```

a) Nusret

b) Nusret Gökçe

c) Nusret Nusret Gökçe

d) compile error

---

13. what's wrong with the following code

```
//method declaration
public static int max(int number1, int number2) {
    if (number1 > number2) {
        return number1;
    } else {
        return number2;
    }
}
public static void main(String[] args) {
    int number1 = 5;
    int number2 = 10;
    int number3 = -15;
    number1 = max(number1, number2, number3);
}
```

a) varible number1 can NOT be used in both side of the assignment oprator witch causes undefined behavior

b) Parametet of the method `max` confilicts with the varibles `number1` and `number2` in the main method

c) if-else stracture is not correct

d) the function must be given two arguments, not three.

---

14. what's missing in the following code

```
//method declaration
public static whaaaat() {
    System.out.println("whaaaat");
}
public static void main(String[] args) {
    whaaaat();
}
```

---

15. how can you fill the blank to make the code work

```
//method declaration
public static double devideDouble(double number1, double number2) {
    return number1 / number2;
}
public static int devideInt(int number1, int number2) {
    return number1 / number2;
}
public static void main(String[] args) {
    double number1 = 5.0;
    double number2 = 2.0;
    int number3 = 5;
    int number4 = 2;
    System.out.println(devideDouble(______, _______));
    System.out.println(devideInt(______, _______));
}
```

Ambiguous method calls

- if there are multiple methods with the same name and the same number of arguments, the compiler cannot determine which method to call
- this is a compile-time error

  Exercises

  16. will the following code show any output. if yes, is it output all the expected output

```
//method declaration
public static int addOne(int number1) {
    return number1 + 1;
}
public static int addOne(int number2) {
    return number2++;
}
public static void main(String[] args) {
    System.out.println("hey i got executed before the method call");
    int number1 = 5;
    int number2 = 10;
    System.out.println(addOne(number1));
    System.out.println(addOne(number2));
}
```

Java Scope

- in java, varibles are only visible in the block in which they are declared
- variables declared in a block are not visible in other blocks
- variables declared in a block are visible in all sub-blocks of that block
- variables declared in a block are not visible in the block that contains that block
- java compiler will not allow you to declare two variables with the same name if one is visible to the other
- blocks in java are methods, if-else, for, while, switch, etc.

  Exercises

17. what's the output of the following code

```
public static void main(String[] args) {
    int number = 5;
    if (number == 5) {
        int number2 = 10;
        System.out.println(number2);
    }
    System.out.println(number2);
}
```

a) 5

b) 10

c) 5 10

d) compile error

---

18. why this code can not be compiled

```
public static void main(String[] args) {
    System.out.println(number);
    int number = 5;
}
```

---

19. fix the following code with minimal changes

```
public static void method1() {
    int number1 = 5;
    System.out.println(number1);
}
public static void method2() {
    int number2 = 10;
    for(int i=0; number2 != 20; number2++) {
        System.out.println(number2);
}
    System.out.println(i);
public static void main(String[] args) {
    method1();
    System.out.println(number1);
    method2();
}
```

---

answers:

1. A
2. D
3. C
4. `printName("Nusret");`

`printName("Gordon");`

```
printName("Guy Fury");
```

5. A
6. Compiler error. parameters should be declared with type
7. Compiler error. parameters should be separated by commas
8. false, true
9. return type of the method should be int
10. A
11. one way is to put the return value in a variable and print it

    ```
    int newNumber = change(number);
    System.out.println(newNumber);
    ```

12. C
13. D
14. return type
15. number1, number2, number3, number4 (left to right, up to down order)
16. no, it will not show any output. it will show a compile error. because the method names are the same and both methods have the same number of arguments. so the compiler cannot determine which method to call.
17. compile error. number2 is not visible in the main method. it is only visible in the if block
18. a declared variable should be initialized before it is used. so the variable `number` should be declared after it is initialized.
19. one way is as follows

    ```
    public static void method1() {
        int number1 = 5;
        System.out.println(number1);
    }
    public static void method2() {
        int number2 = 10;
        for(int i=0; i < 10; i++) {
            System.out.println(number2);
        }
        System.out.println(number2);
    }
    public static void main(String[] args) {
        method1();
        int number1 = 5;
        System.out.println(number1);
        method2();
    }
    ```