CS102A Introduction to Computer Programming
Fall 2020 Lab 3

Objective

1. Learn how to obtain user input from command line arguments or by the `Scanner` class.
2. Learn how to use the `if` and `if`...`else` selection statements to choose between alternative actions.
3. Learn how to use the `while` repetition statement to execute statements in a program repeatedly.

1 Exercise

1.1 Exercise 1

If we run a Java Program by using the command `java Hello I Love Programming` where the name of the class is `Hello`, then it will run `Hello.class`. For the string after the class name `Hello`, i.e, `I Love Programming`, these are *command line arguments*.

When command line arguments are supplied to JVM, JVM wraps these and supply to `String[] args` in your `main` method. It can be confirmed that they are actually wrapped up in args array by checking the length of `args` using `args.length`.

Write an application to get user's name, age, weight and grade through command line arguments, then prints out in a specific format. It should have similar output as Exercise 2 in Lab 2.

```java
public class Lab3_E1 {
    public static void main(String[] args) {
        String name = args[0];
        int age = Integer.parseInt(args[1]);
        float weight = Float.parseFloat(args[2]);
        char grade = args[3].charAt(0);

        System.out.printf("You are %s.\nYou are %d years old.\n", name, age);
        System.out.printf("You weigh %.1f KG.\nThe highest grade you got is %c.\n", weight, grade);
    }
}
```

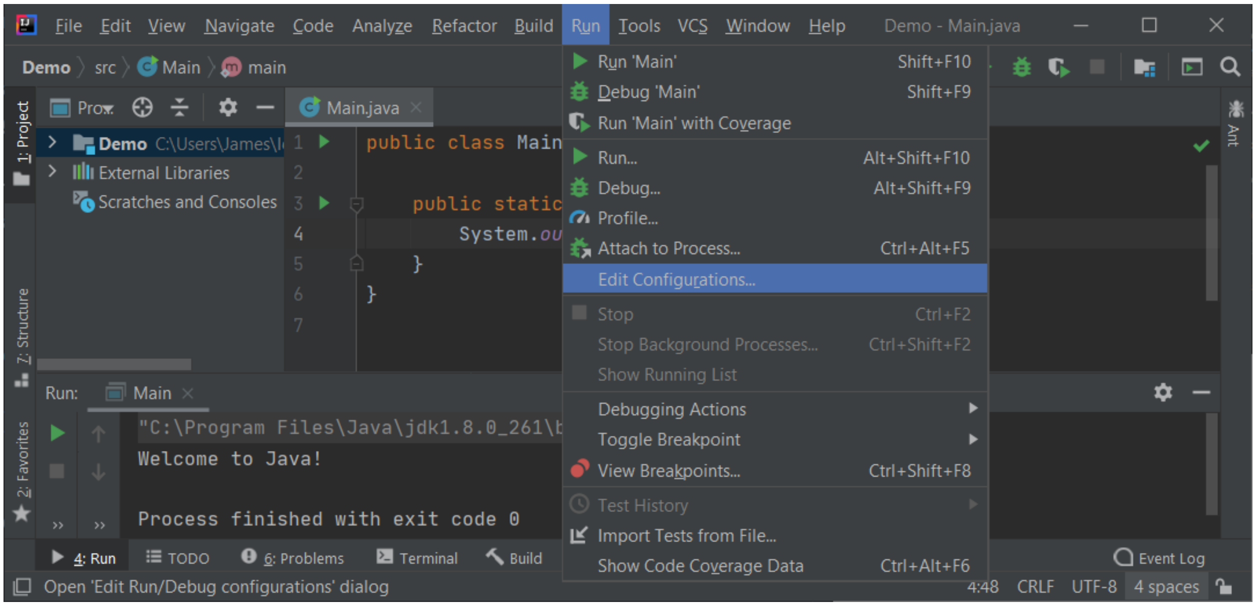
Command line arguments can be obtained in IDE or in command line:

1. in command line:

```
> javac Lab3_E1.java

> java Lab3_E1 zhangsan 20 60.1 A
You are zhangsan.
You are 20 years old.
You weigh 60.1 KG.
The highest grade you got is A.
```

2. in IDEA: There is an editbox labeled *Program arguments* in the *Configuration* tab of project settings. Input `zhangsan 20 60.1 A` in the box.



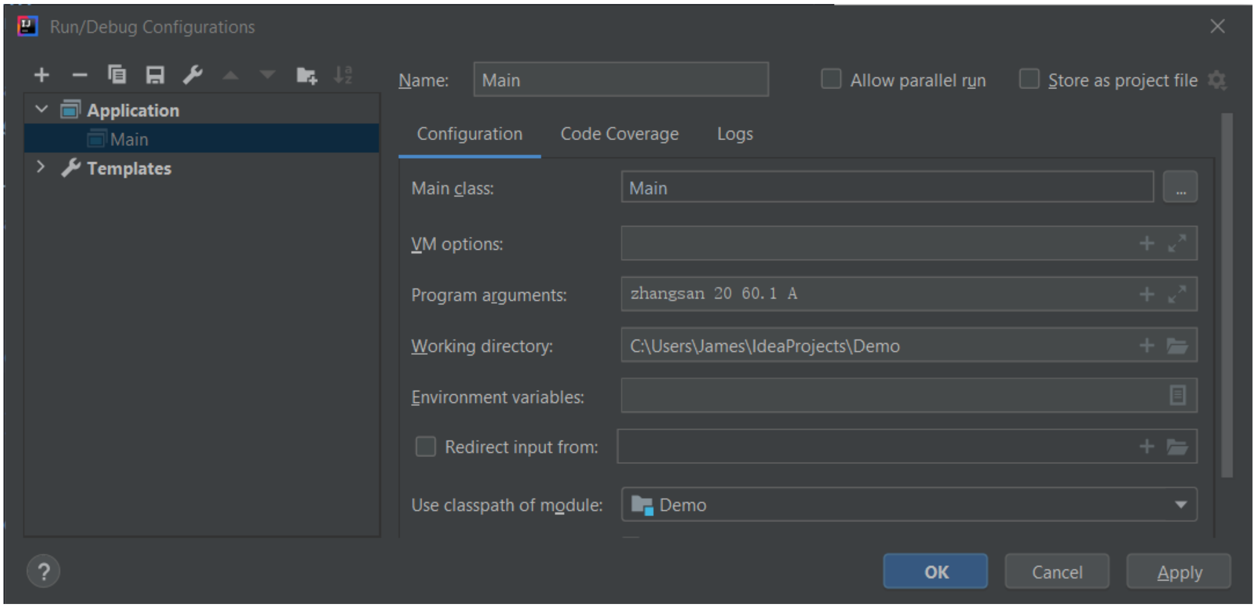
```java
public class Main

    public static
        System.ou
    }
}
```

```
"C:\Program Files\Java\jdk1.8.0_261\b
Welcome to Java!

Process finished with exit code 0
```

Open 'Edit Run/Debug configurations' dialog



Name: Main

☐ Allow parallel run

☐ Store as project file

Main class: Main

VM options:

Program arguments: zhangsan 20 60.1 A

Working directory: C:\Users\James\IdeaProjects\Demo

Environment variables:

☐ Redirect input from:

Use classpath of module: Demo

? Question

What will happen if you input more or less than four parameters, or input wrong data type?

1.2 Exercise 2

Write an application which can convert the grades on 100 point scale into GPA according to the following table.

Grade	GPA
100–90	4.0
89–80	3.0
79–70	2.0
69–60	1.0
59–0	0

Sample code:

```
public class Lab3_E2 {
    public static void main(String[] args) {
        float gpa;
        float score = Float.parseFloat(args[0]);

        if(score >= 60)
            System.out.println("You passed the exam.");
```

```java
        else
            System.out.println("You failed in the exam.");

        if(score >= 90)
            gpa = (float) 4.0;
        else if(score >= 80)
            gpa = 3.0f;
        else if(score >= 70)
            gpa = 2.0f;
        else if(score >= 60)
            gpa = 1.0f;
        else if(score >  0)
            gpa = 0.0f;
        else {
            System.out.println("Invalide grade");
            return;
        }
        System.out.printf("Your score is %.1f, the GPA is %.1f\n", score, gpa);
    }
}
```

Sample output:

```
> javac Lab3_E2.java

> java Lab3_E2 96
You passed the exam.
Your score is 96.0, the GPA is 4.0

> java Lab3_E2 85
You passed the exam.
Your score is 85.0, the GPA is 3.0

> java Lab3_E2 77
You passed the exam.
Your score is 77.0, the GPA is 2.0

> java Lab3_E2 60
```

```
You passed the exam.
Your score is 60.0, the GPA is 1.0

> java Lab3_E2 59
You failed in the exam.
Your score is 59.0, the GPA is 0.0
```

1.3 Exercise 3

Write a program to print 9×9 multiplication table, by using the while repetition statement.

Note
Use `printf` with `%02d`, `3` will become `03`; with `%2d`, `3` will become ` 3` with a space on the left.

Sample output:

```
1 * 1 =  1
1 * 2 =  2  2 * 2 =  4
1 * 3 =  3  2 * 3 =  6  3 * 3 =  9
1 * 4 =  4  2 * 4 =  8  3 * 4 = 12  4 * 4 = 16
1 * 5 =  5  2 * 5 = 10  3 * 5 = 15  4 * 5 = 20  5 * 5 = 25
1 * 6 =  6  2 * 6 = 12  3 * 6 = 18  4 * 6 = 24  5 * 6 = 30  6 * 6 = 36
1 * 7 =  7  2 * 7 = 14  3 * 7 = 21  4 * 7 = 28  5 * 7 = 35  6 * 7 = 42  7
* 7 = 49
1 * 8 =  8  2 * 8 = 16  3 * 8 = 24  4 * 8 = 32  5 * 8 = 40  6 * 8 = 48  7
* 8 = 56  8 * 8 = 64
1 * 9 =  9  2 * 9 = 18  3 * 9 = 27  4 * 9 = 36  5 * 9 = 45  6 * 9 = 54  7
* 9 = 63  8 * 9 = 72  9 * 9 = 81
```

1.4 Exercise 4

Create a class called `GuessingNumber`. In the `main` method, you should generate a random integer `magicNum` between 0 and 9, then keep asking the user to input an integer between 0 and 9 until the input number is equal to the attribute `magicNum`. When the input number is greater than the attribute `magicNum`, the method should output *Too big! Please try again:*.When the input number is less than the attribute `magicNum`, the method should output *Too small! Please try again:*. Then the method wait for the user to input a new integer. When the input number is equal to the attribute `magicNum`, the method should output *Congratulations!* and terminate.

Sample code:

```java
import java.util.Random;
import java.util.Scanner;

public class GuessingNumber {
    public static void main(String[] args) {
        Random random = new Random();
        int magicNum = random.nextInt(10);

        int inputNum;

        Scanner sc = new Scanner(System.in);

        System.out.println("Please input an Integer in {0,1,2,...,9}:");
        inputNum = sc.nextInt();

        while (                    ){ // Finish this line
            if (                   )  // Finish this line
                System.out.println("Too big! Please try again:");
            else
                System.out.println("Too small! Please try again:");
            inputNum = sc.nextInt();
        }
        System.out.println("Congratulations!");
        sc.close();
    }
}
```

Sample output:

```
Please input an Integer in {0,1,2,...,9}:
1
Too small! Please try again:
5
Too small! Please try again:
8
Too small! Please try again:
9
Congratulations!
```

1.5 Exercise 5

Calculate the value of π from the infinite series $\pi = 4 - 4/3 + 4/5 - 4/7 + 4/9 - 4/11 + \cdots$

Input an integer `n` which represents the number of terms in the formula above. It is more precise when `n` is bigger. Use `do`...`while` or `while` repetition statements to compute the value of π.

Sample output:

```
Please input the precision:
10000
The estimation of Pi is 3.141498
```

Modify your program as follows: Input a double value which represents a precision threshold. Your program should terminate when the difference between two successive iterations is smaller than the precision threshold. Print the value of π, and the iteration numbers. Sample output:

```
Please input the precision:
0.0001
The estimation of Pi is 3.141547
It computed 19998 times
```

Tip

Use `Math.abs()`.