

Q1 Number 1

```
package Assignment1;

import java.util.ArrayList;
import java.util.Random;

public class Q1_SeqV {

    public static ArrayList<STUDENT> students = new ArrayList<STUDENT>();

    public static float getAveGpa(ArrayList<STUDENT> studentsParam) {

        float totalGpa = 0;

        for (STUDENT student : studentsParam) {

            totalGpa += student.getGpa();

        }

        return totalGpa/studentsParam.size();

    }

    public static int getAveAge(ArrayList<STUDENT> studentsParam) {

        int totalAge = 0;

        for (STUDENT student : studentsParam) {

            totalAge += student.getGpa();

        }

        return totalAge/studentsParam.size();

    }

}
```

```
public static void main(String[] args) {  
  
    //init start time  
  
    long startTime = System.nanoTime();  
  
  
    //create 10000 student and store in a list  
  
    for (int i = 0; i < 10000; i++) {  
  
        float gpa = new Random().nextFloat() * (4.0f - 2.0f) + 2.0f;  
  
        int age = (int)(Math.random() * (30-18)) + 18;  
  
        STUDENT student = new STUDENT(gpa, age);  
  
        students.add(student);  
  
    }  
  
  
    //get the average gpa  
  
    float aveGpa = getAveGpa(students);  
  
  
    //get the average age  
  
    int aveAge = getAveAge(students);  
  
  
    //calculate the execution time  
  
    long endTime = System.nanoTime();  
  
    long second = (endTime - startTime)/100000000;  
  
    long milisecond = (endTime - startTime)/10000;  
  
  
    System.out.println("Created a list of 10,000 students in sequential version");  
  
    System.out.println("Average GPA is: " + aveGpa);  
}
```

```

        System.out.println("Average Age is: " + aveAge);

        System.out.println("This excution take: " + + second + "." + milisecond + "s");

    }

}

```

```

Created a list of 10,000 students in sequential version
Average GPA is: 3.0029218
Average Age is: 2
This excution take: 0.701s

```

Q1 Number 2

```

package Assignment1;

import java.util.ArrayList;
import java.util.Random;

public class Q1_PLV1 {

    public static ArrayList<STUDENT> students1 = new ArrayList<STUDENT>();

    public static ArrayList<STUDENT> students2 = new ArrayList<STUDENT>();

    static float aveGpa1 = 0;

    static int aveAge1 = 0;

    static float aveGpa2 = 0;

    static int aveAge2 = 0;

```

```
public static float getAveGpa(ArrayList<STUDENT> studentsParam) {  
    float totalGpa = 0;  
    for (STUDENT student : studentsParam) {  
        totalGpa += student.getGpa();  
    }  
    return totalGpa/studentsParam.size();  
}
```

```
public static int getAveAge(ArrayList<STUDENT> studentsParam) {  
    int totalAge = 0;  
    for (STUDENT student : studentsParam) {  
        totalAge += student.getGpa();  
    }  
    return totalAge/studentsParam.size();  
}
```

```
public static void func1() {  
    //create 5000 student and store in a list  
    for (int i = 0; i < 5000; i++) {  
        float gpa = new Random().nextFloat() * (4.0f - 2.0f) + 2.0f;  
        int age = (int)(Math.random() * (30-18)) + 18;  
        STUDENT student = new STUDENT(gpa, age);  
        students1.add(student);  
    }  
}
```

```

        //get average gpa and age

        aveGpa1 = getAveGpa(students1);

        aveAge1 = getAveAge(students1);

    }

    public static void func2() {

        //create 5000 student and store in a list

        for (int i = 0; i < 5000; i++) {

            float gpa = new Random().nextFloat() * (4.0f - 2.0f) + 2.0f;

            int age = (int)(Math.random() * (30-18)) + 18;

            STUDENT student = new STUDENT(gpa, age);

            students2.add(student);

        }

        //get average gpa and age

        aveGpa2 = getAveGpa(students2);

        aveAge2 = getAveAge(students2);

    }

    public static void main(String[] args) {

        Thread thread1 = new Thread(

            new Runnable() {

                public void run() {

                    func1();

```

```

        }

    });

    Thread thread2 = new Thread(
        new Runnable() {
            public void run() {
                func2();
            }
        });

    long startTime = System.nanoTime();

    thread1.start();
    thread2.start();

    try {
        thread1.join();
        thread2.join();
    } catch (InterruptedException e) {
        e.printStackTrace();
    }

    //get the final average gpa and age
    float aveGpaAll = (aveGpa1 + aveAge2) / 2;
    int aveAgeAll = (aveAge1 + aveAge2) / 2;

```

```

        //calculate execution time

        long endTime = System.nanoTime();

        long second = (endTime - startTime)/100000000;

        long milisecond = (endTime - startTime)/10000;


        System.out.println("Created 2 list of 5,000 students in parallel version 1");

        System.out.println("Average GPA is: " + aveGpaAll);

        System.out.println("Average Age is: " + aveAgeAll);

        System.out.println("This excution take: " + + second + "." + milisecond + "s");

    }

}

Created 2 list of 5,000 students in parallel version 1
Average GPA is: 2.4963684
Average Age is: 2
This excution take: 0.832s

```

Q1 Number 3

```

package Assignment1;


import java.util.ArrayList;

import java.util.Random;

```

```
public class Q1_PLV2 {

    public static ArrayList<STUDENT> students1 = new ArrayList<STUDENT>();

    public static ArrayList<STUDENT> students2 = new ArrayList<STUDENT>();

    public static ArrayList<STUDENT> students3 = new ArrayList<STUDENT>();

    public static ArrayList<STUDENT> students4 = new ArrayList<STUDENT>();

    public static ArrayList<STUDENT> students5 = new ArrayList<STUDENT>();

    static float aveGpa1 = 0;

    static int aveAge1 = 0;

    static float aveGpa2 = 0;

    static int aveAge2 = 0;

    static float aveGpa3 = 0;

    static int aveAge3 = 0;

    static float aveGpa4 = 0;

    static int aveAge4 = 0;

    static float aveGpa5 = 0;

    static int aveAge5 = 0;

    public static float getAveGpa(ArrayList<STUDENT> studentsParam) {

        float totalGpa = 0;

        for (STUDENT student : studentsParam) {

            totalGpa += student.getGpa();

        }

        return totalGpa/studentsParam.size();

    }

}
```



```
public static int getAveAge(ArrayList<STUDENT> studentsParam) {  
  
    int totalAge = 0;  
  
    for (STUDENT student : studentsParam) {  
  
        totalAge += student.getGpa();  
  
    }  
  
    return totalAge/studentsParam.size();  
  
}
```

```
public static void func1() {  
  
    //create 2000 student and store in a list  
  
    for (int i = 0; i < 2000; i++) {  
  
        float gpa = new Random().nextFloat() * (4.0f - 2.0f) + 2.0f;  
  
        int age = (int)(Math.random() * (30-18)) + 18;  
  
        STUDENT student = new STUDENT(gpa, age);  
  
        students1.add(student);  
  
    }  
  
  
    //get average gpa and age  
  
    aveGpa1 = getAveGpa(students1);  
  
    aveAge1 = getAveAge(students1);  
  
}
```

```
public static void func2() {  
  
    //create 2000 student and store in a list  
  
    for (int i = 0; i < 2000; i++) {
```

```

        float gpa = new Random().nextFloat() * (4.0f - 2.0f) + 2.0f;

        int age = (int)(Math.random() * (30-18)) + 18;

        STUDENT student = new STUDENT(gpa, age);

        students2.add(student);

    }

    //get average gpa and age

    aveGpa2 = getAveGpa(students2);

    aveAge2 = getAveAge(students2);

}

public static void func3() {

    //create 2000 student and store in a list

    for (int i = 0; i < 2000; i++) {

        float gpa = new Random().nextFloat() * (4.0f - 2.0f) + 2.0f;

        int age = (int)(Math.random() * (30-18)) + 18;

        STUDENT student = new STUDENT(gpa, age);

        students3.add(student);

    }

    //get average gpa and age

    aveGpa3 = getAveGpa(students3);

    aveAge3 = getAveAge(students3);

}

```

```
public static void func4() {  
  
    //create 2000 student and store in a list  
  
    for (int i = 0; i < 2000; i++) {  
  
        float gpa = new Random().nextFloat() * (4.0f - 2.0f) + 2.0f;  
  
        int age = (int)(Math.random() * (30-18)) + 18;  
  
        STUDENT student = new STUDENT(gpa, age);  
  
        students4.add(student);  
  
    }  
  
  
    //get average gpa and age  
  
    aveGpa4 = getAveGpa(students4);  
  
    aveAge4 = getAveAge(students4);  
  
}
```

```
public static void func5() {  
  
    //create 2000 student and store in a list  
  
    for (int i = 0; i < 2000; i++) {  
  
        float gpa = new Random().nextFloat() * (4.0f - 2.0f) + 2.0f;  
  
        int age = (int)(Math.random() * (30-18)) + 18;  
  
        STUDENT student = new STUDENT(gpa, age);  
  
        students5.add(student);  
  
    }  
  
  
    //get average gpa and age  
  
    aveGpa5 = getAveGpa(students5);
```

```
        aveAge5 = getAveAge(students5);  
    }  
}
```

```
public static void main(String[] args) {  
    Thread thread1 = new Thread(  
        new Runnable() {  
            public void run() {  
                func1();  
            }  
        });  
}
```

```
    Thread thread2 = new Thread(  
        new Runnable() {  
            public void run() {  
                func2();  
            }  
        });  
}
```

```
    Thread thread3 = new Thread(  
        new Runnable() {  
            public void run() {  
                func3();  
            }  
        });  
}
```

```
Thread thread4 = new Thread(  
    new Runnable() {  
        public void run() {  
            func4();  
        }  
    });
```

```
Thread thread5 = new Thread(  
    new Runnable() {  
        public void run() {  
            func5();  
        }  
    });
```

```
long startTime = System.nanoTime();
```

```
thread1.start();
```

```
thread2.start();
```

```
thread3.start();
```

```
thread4.start();
```

```
thread5.start();
```

```
try {  
    thread1.join();  
    thread2.join();
```

```
        thread3.join();

        thread4.join();

        thread5.join();
    } catch (InterruptedException e) {

        e.printStackTrace();
    }
}
```

```
//get the final average gpa and age
```

```
float aveGpaAll = (aveGpa1 + aveAge2 + aveAge3 + aveAge4 + aveAge5) / 5;
```

```
int aveAgeAll = (aveAge1 + aveAge2 + aveAge3 + aveAge4 + aveAge5) / 5;
```

```
//calculate execution time
```

```
long endTime = System.nanoTime();
```

```
long second = (endTime - startTime)/100000000;
```

```
long milisecond = (endTime - startTime)/10000;
```

```
System.out.println("Created 5 list of 2,000 students in parallel version 1");
```

```
System.out.println("Average GPA is: " + aveGpaAll);
```

```
System.out.println("Average Age is: " + aveAgeAll);
```

```
System.out.println("This excution take: " + + second + "." + milisecond + "s");
```

```
}
```

}

Created 5 list of 2,000 students in parallel version 1
Average GPA is: 2.1956096
Average Age is: 2
This excution take: 0.776s