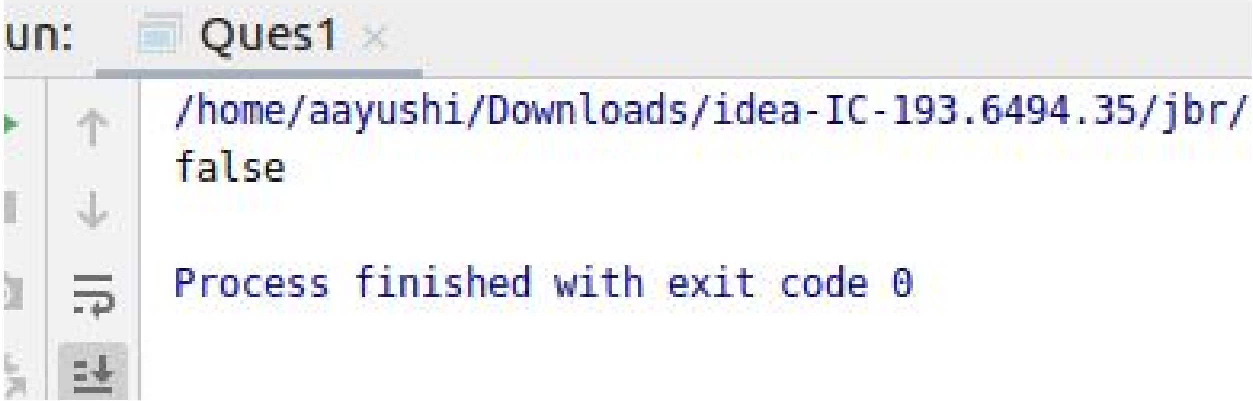
Session : Java 8 Features

1. **Write the following a functional interface and implement it using lambda:**
   - ○ **(1)First number is greater than second number or not Parameter (int ,int) Return boolean**

CODE

```
package aayushi;

interface Ques1Interface{
  public boolean greater(int a,int b);
}

public class Ques1 {
  public static void main(String[] args) {

       Ques1Interface obj = (a,b) -> {
             return (a>b) ? true:false;
          };

       System.out.println(obj.greater(2,3));
  }
}
```

OUTPUT



```
/home/aayushi/Downloads/idea-IC-193.6494.35/jbr/
false

Process finished with exit code 0
```

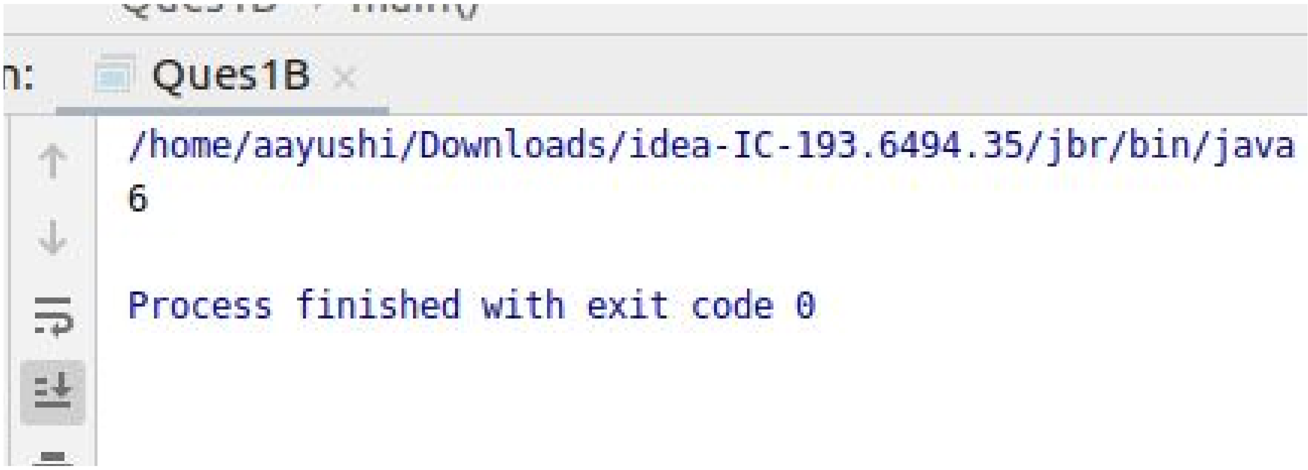
(2) Increment the number by 1 and return incremented value Parameter (int) Return int

CODE

```
package aayushi;
```

```
interface Ques1BInterface{
  public int increment(int a);
}

public class Ques1B {
  public static void main(String[] args) {

    Ques1BInterface obj = (a)-> {
        return ++a;
      };

    System.out.println(obj.increment(5));
  }
}
```

OUTPUT



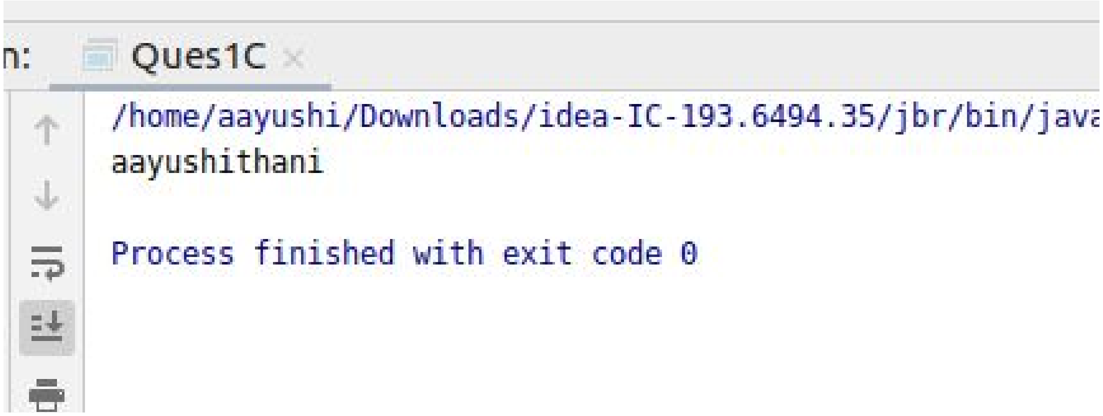
```
/home/aayushi/Downloads/idea-IC-193.6494.35/jbr/bin/java
6

Process finished with exit code 0
```

(3) Concatination of 2 string   Parameter (String , String) Return (String)

CODE

```
package aayushi;

interface Ques1CInterface{
  public String concat(String x,String y);
}

public class Ques1C {
  public static void main(String[] args) {
    Ques1CInterface obj = (x,y) -> {
        return x+y;
        };

        System.out.println(obj.concat("aayushi","thani"));
    }
}
```

OUTPUT



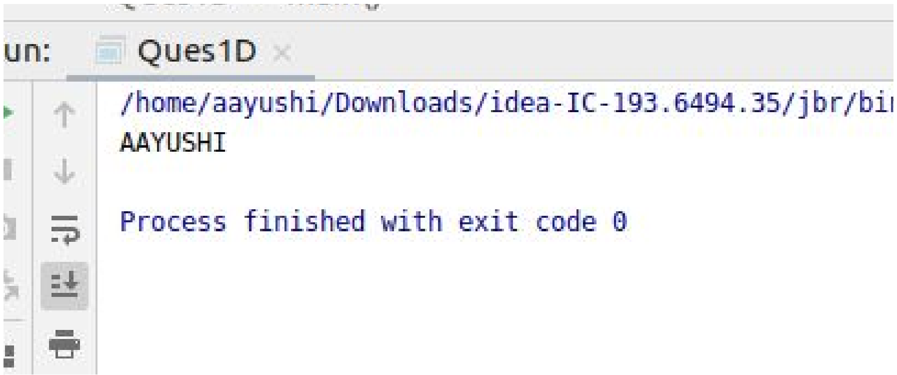
```
/home/aayushi/Downloads/idea-IC-193.6494.35/jbr/bin/java
aayushithani

Process finished with exit code 0
```

(4) Convert a string to uppercase and return . **Parameter (String) Return (String)**

CODE

```
package aayushi;

interface Ques1DInterface{
    public String uppercase(String str);
}

public class Ques1D {
    public static void main(String[] args) {

        Ques1DInterface obj = (str) -> {
            return str.toUpperCase();
        };

        System.out.println(obj.uppercase("aayushi"));
    }
}
```

OUTPUT



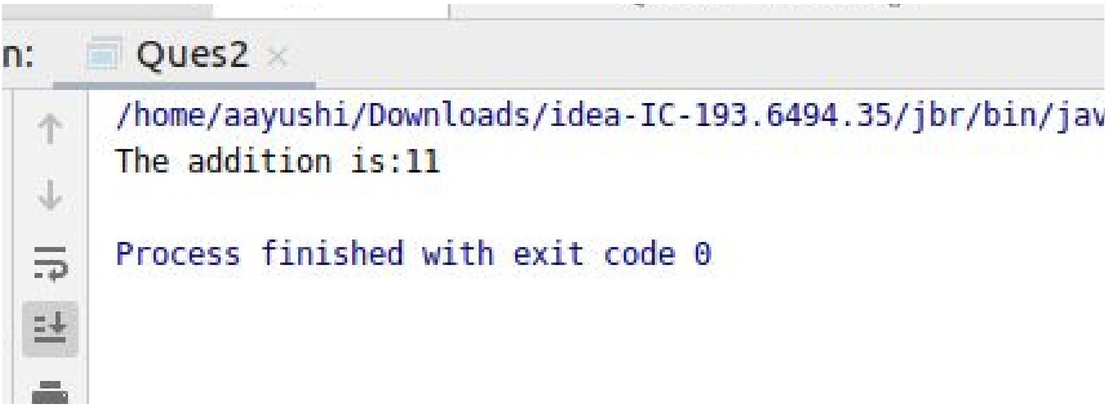
```
/home/aayushi/Downloads/idea-IC-193.6494.35/jbr/bi
AAYUSHI

Process finished with exit code 0
```

2. **Create a functional interface whose method takes 2 integers and return one integer.**

   CODE

```
package aayushi;

@FunctionalInterface
interface Ques2Interface{
   public int add(int x, int y);
}

public class Ques2 {
   public static void main(String[] args) {

       Ques2Interface obj = (x,y) -> {
              return x+y;
       };

       System.out.println("The addition is:"+ obj.add(5,6));
   }
}
```

OUTPUT



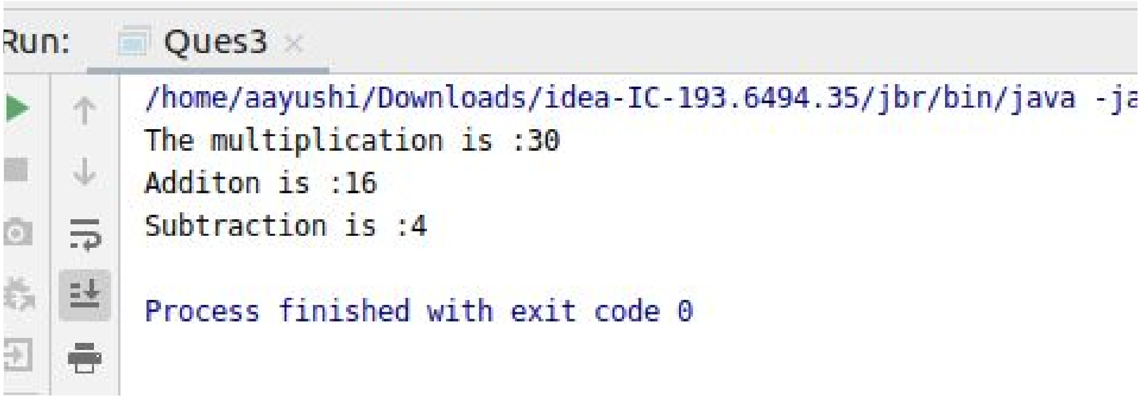
```
/home/aayushi/Downloads/idea-IC-193.6494.35/jbr/bin/jav
The addition is:11

Process finished with exit code 0
```

3. **Using (instance) Method reference create and apply add and subtract method and using (Static) Method reference create and apply multiplication method for the functional interface created.**

CODE

```
package aayushi;

interface Ques3Interface{
  public void something(int x , int y);
}

public class Ques3 {
  public static void multiply(int x , int y)
  {
     System.out.println("The multiplication is :" + (x*y));
  }


  public void addsub(int x, int y)
  {
     System.out.println("Additon is :"+ (x+y));
     System.out.println("Subtraction is :"+ (x-y));
  }


  public static void main(String[] args) {
     Ques3Interface operate1 = Ques3::multiply; //(Static) Method reference
     operate1.something(5,6);

     Ques3 methodReference = new Ques3(); // (instance) Method reference
     Ques3Interface operate2 = methodReference::addsub;
     operate2.something(10,6);
  }
}
```

OUTPUT



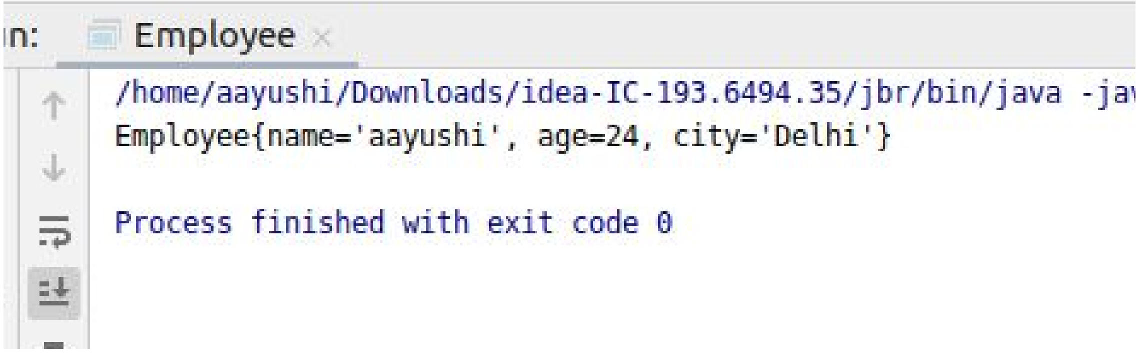
```
/home/aayushi/Downloads/idea-IC-193.6494.35/jbr/bin/java -ja
The multiplication is :30
Additon is :16
Subtraction is :4

Process finished with exit code 0
```

4. **Create an Employee Class with instance variables (String) name, (Integer)age, (String)city and get the instance of the Class using constructor reference.**

CODE

```
package aayushi;

interface ConstInterface{
  Employee ConstMethod(String name,Integer age, String city);
}

class Employee{
  String name;
  Integer age;
  String city;

  Employee(String name, Integer age, String city)
  {
     this.name = name;
     this.age = age;
     this.city= city;
  }

  @Override
  public String toString() {
     return "Employee{" +
           "name='" + name + '\'' +
                ", age=" + age +
                ", city='" + city + '\'' +
                '}';
    }

    public static void main(String[] args) {
        ConstInterface constref = Employee::new;
        System.out.println(constref.ConstMethod("aayushi", 24, "Delhi"));
    }
}
```

<u>OUTPUT</u>



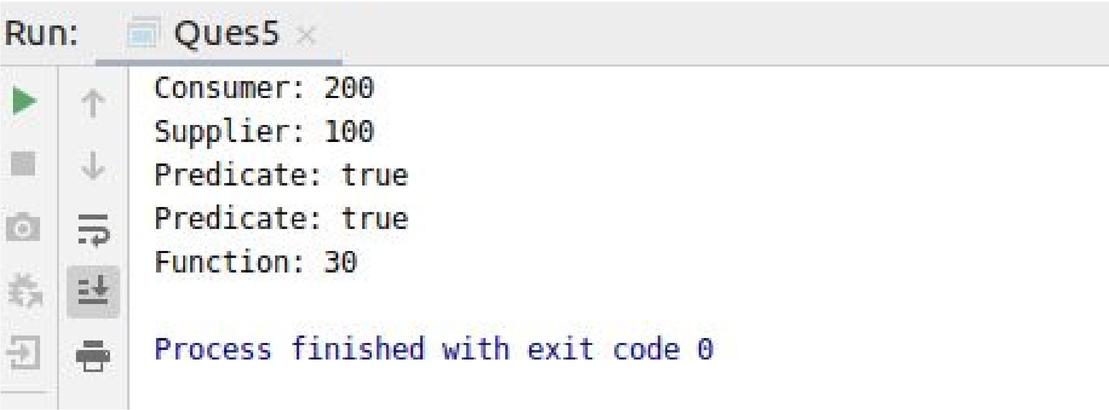
```
/home/aayushi/Downloads/idea-IC-193.6494.35/jbr/bin/java -ja
Employee{name='aayushi', age=24, city='Delhi'}

Process finished with exit code 0
```

5. **Implement following functional interfaces from java.util.function using lambdas:**
    - ○ **(1) Consumer**
    - ○ **(2) Supplier**
    - ○ **(3) Predicate**
    - ○ **(4) Function**

<u>CODE</u>

```
package aayushi;

import java.util.function.BiFunction;
import java.util.function.Consumer;
import java.util.function.Predicate;
import java.util.function.Supplier;

public class Ques5 {
    public static void main(String[] args) {

        //CONSUMER FUNCTIONAL INTERFACE
        Consumer<Integer> consumer = (e1) -> {
            System.out.println("Consumer: " + e1);
        };
        consumer.accept(200);

        //SUPPLIER FUNCTIONAL INTERFACE
        Supplier<Integer> supplier = () -> {
            System.out.print("Supplier: ");
            return 100;
        };
        System.out.println(supplier.get());

        //PREDICATE FUNCTIONAL INTERFACE
        Predicate<Integer> predicate = (e) -> e%2==0; //even --returns boolean value
        System.out.println("Predicate: "+predicate.test(4));

        Predicate<String> predicate1 = (e) -> e.startsWith("no"); //checks whether the string starts
with no.
        System.out.println("Predicate: "+predicate1.test("nope"));

        //FUNCTION FUNCTIONAL INTERFACE
        BiFunction<Integer,Long,Integer> bifun=(e, e1)-> e+ Integer.parseInt(e1.toString());
        System.out.println("Function: "+bifun.apply(10,20L));  //add
    }

}
```

<u>OUTPUT</u>



```
Consumer: 200
Supplier: 100
Predicate: true
Predicate: true
Function: 30

Process finished with exit code 0
```

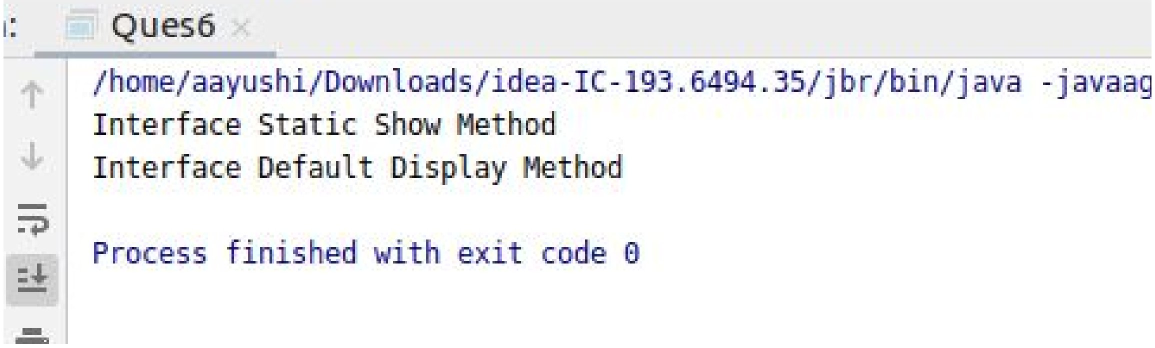
6. **Create and access default and static method of an interface.**

<u>CODE</u>

```
package aayushi;
```

```
interface Ques6interface{
   default void display(){
      System.out.println("Interface Default Display Method");
   }

   static void show(){
      System.out.println("Interface Static Show Method");
   }
}

public class Ques6 implements Ques6interface{
   public static void main(String[] args) {
      Ques6interface.show();
      Ques6 obj = new Ques6();
      obj.display();
   }
}
```

OUTPUT



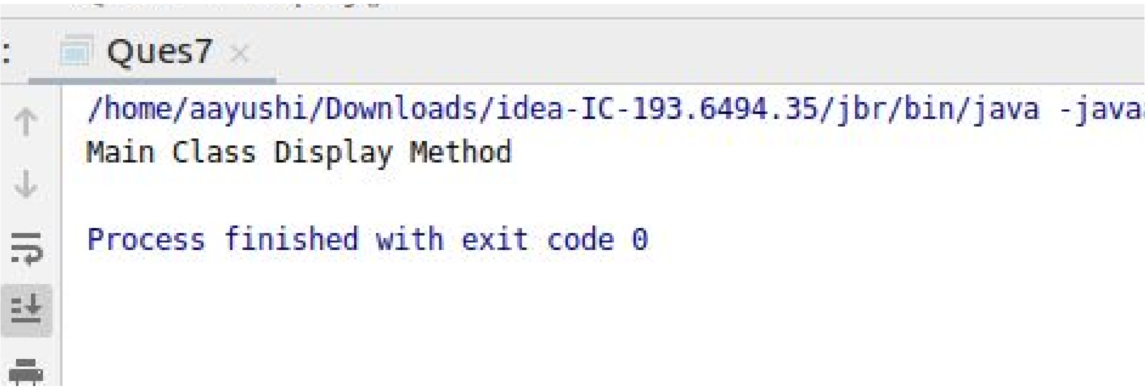
```
/home/aayushi/Downloads/idea-IC-193.6494.35/jbr/bin/java -javaag
Interface Static Show Method
Interface Default Display Method

Process finished with exit code 0
```

7. **Override the default method of the interface.**

CODE

```
package aayushi;

interface NewDefaultInterface
{
   default void display(){
      System.out.println("Interface Default Method");
   }
}

public class Ques7 implements NewDefaultInterface{
   public static void main(String[] args) {
      Ques7 obj = new Ques7();
      obj.display();
   }

   @Override
   public void display() {
      System.out.println("Main Class Display Method");
   }
}
```

OUTPUT



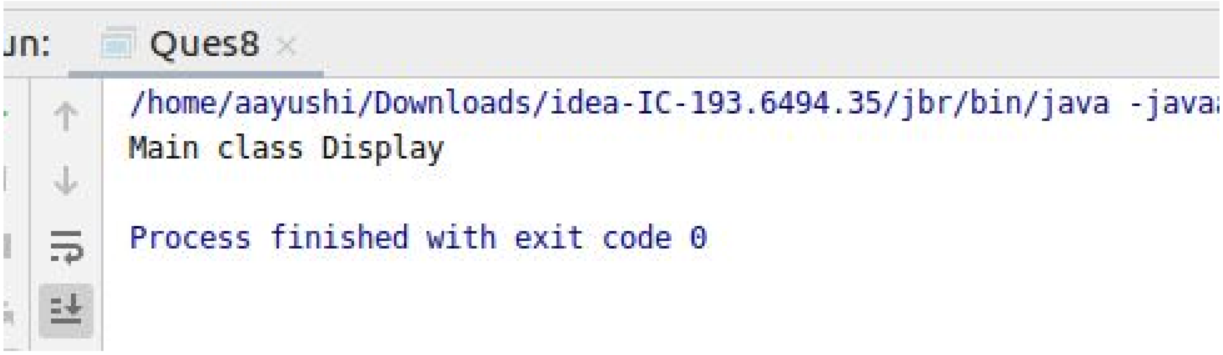
```
/home/aayushi/Downloads/idea-IC-193.6494.35/jbr/bin/java -java
Main Class Display Method

Process finished with exit code 0
```

8. Implement multiple inheritance with default method inside interface.

CODE

```
package aayushi;

interface Defaultinterface{
  default void display(){
     System.out.println("Default Interface 1");
  }
}

interface Defaultinterface1{
  default void display(){
     System.out.println("Default Interface 2");
  }
}

public class Ques8 implements Defaultinterface,Defaultinterface1{
  public static void main(String[] args) {
        Ques8 obj = new Ques8();
        obj.display();
    }

    @Override
    public void display() {
        System.out.println("Main class Display");
    }
}
```

OUTPUT



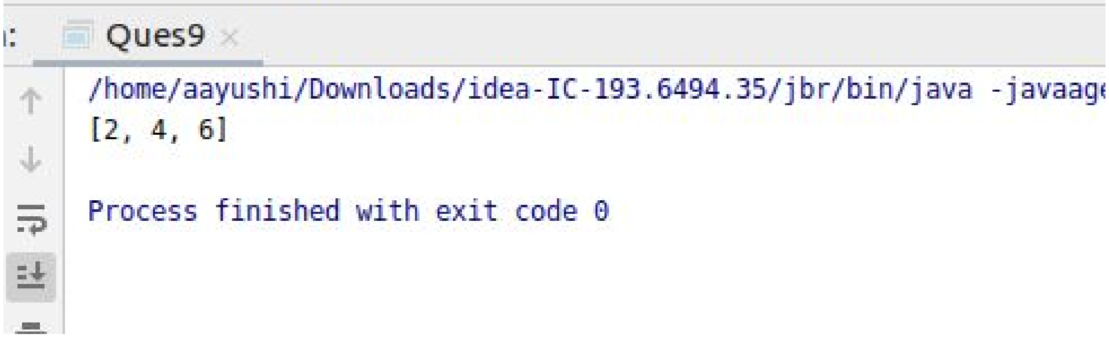
```
/home/aayushi/Downloads/idea-IC-193.6494.35/jbr/bin/java -java
Main class Display

Process finished with exit code 0
```

9. **Collect all the even numbers from an integer list.**

CODE

```
package aayushi;

import java.util.Arrays;
import java.util.stream.Collector;
import java.util.stream.Collectors;

public class Ques9 {
    public static void main(String[] args) {
        System.out.println(
        Arrays.asList(1,2,3,4,5,6)
                .stream()
                .filter(e->e%2==0)
                .collect(Collectors.toList())
        );
    }
}
```

OUTPUT



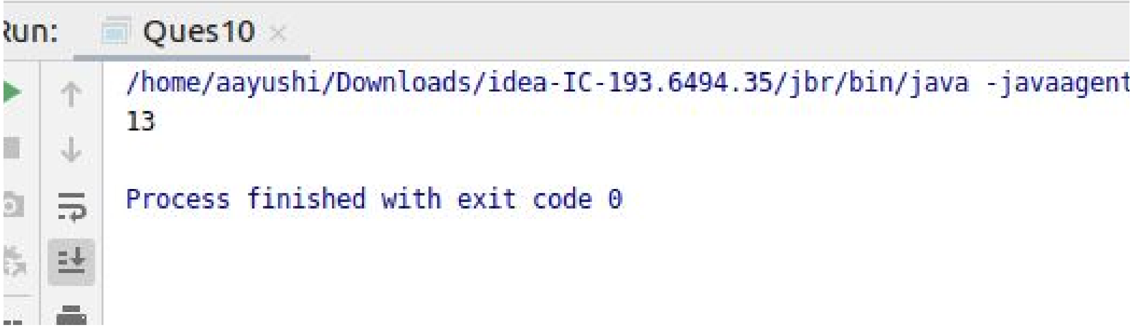
```
/home/aayushi/Downloads/idea-IC-193.6494.35/jbr/bin/java -javaage
[2, 4, 6]

Process finished with exit code 0
```

10. Sum all the numbers greater than 5 in the integer list.

CODE

```
package aayushi;

import java.util.Arrays;

public class Ques10 {
  public static void main(String[] args) {
      System.out.println(
      Arrays.asList(1,2,3,4,5,6,7)
            .stream()
            .filter(e->e>5)
            .mapToInt(e->e)
            .sum()
      );
  }
}
```

OUTPUT



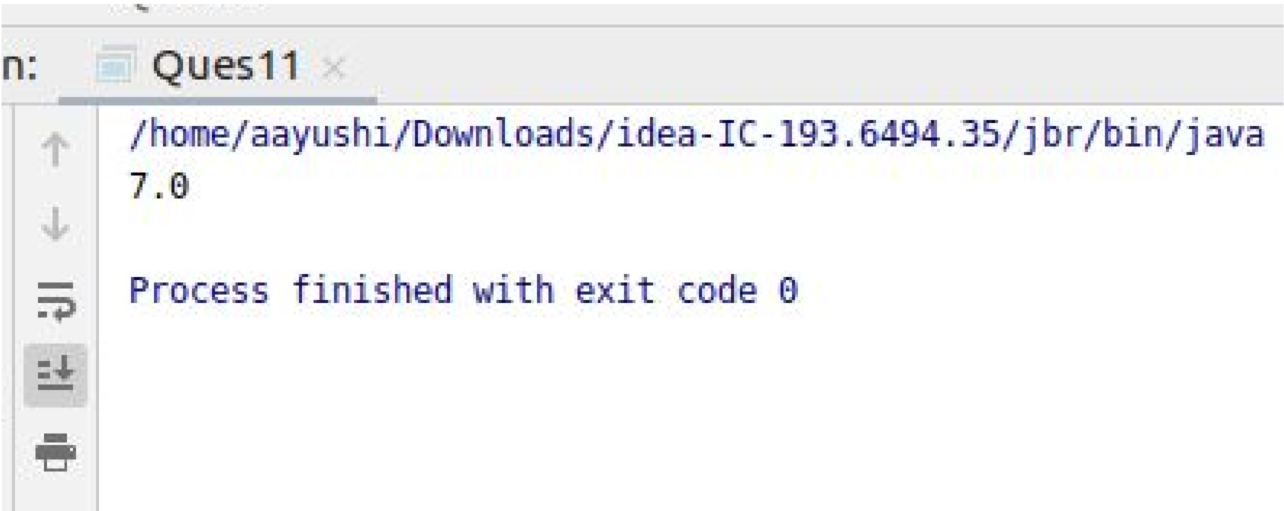
```
/home/aayushi/Downloads/idea-IC-193.6494.35/jbr/bin/java -javaagent
13

Process finished with exit code 0
```

11. Find average of the number inside integer list after doubling it.

CODE

```
package aayushi;
import java.util.Arrays;

public class Ques11 {
  public static void main(String[] args) {
      System.out.println(
      Arrays.asList(1,2,3,4,5,6)
              .stream()
              .mapToInt(e->e*2)
              .average().orElse(-1)
      );

  }
}
```

OUTPUT



```
/home/aayushi/Downloads/idea-IC-193.6494.35/jbr/bin/java
7.0

Process finished with exit code 0
```

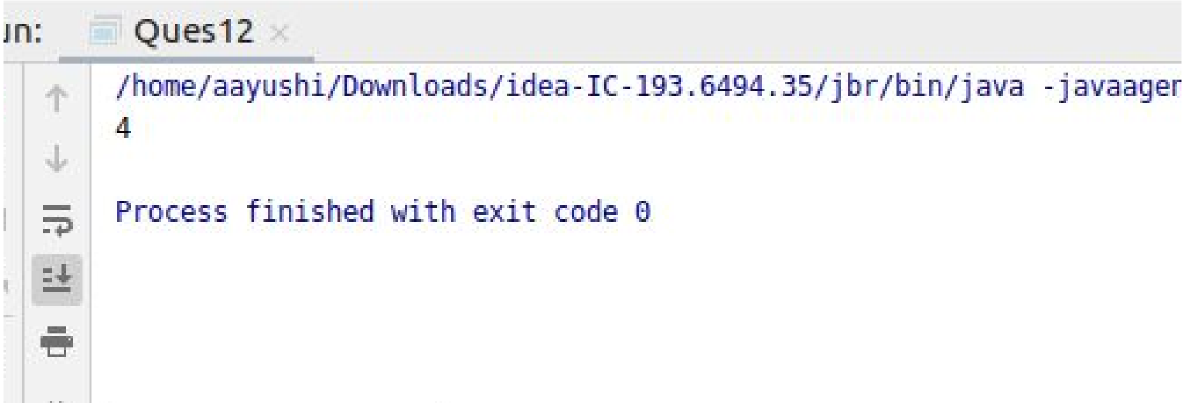
12. Find the first even number in the integer list which is greater than 3.

CODE

```
package aayushi;

import java.util.Arrays;

//Find the first even number in the integer list which is greater than 3.
public class Ques12 {

  public static void main(String[] args) {
      System.out.println(
      Arrays.asList(1,2,3,4,5,6)
```

```
                .stream()
                .filter(e->e%2==0)
                .filter(e->e>3)
                .findFirst().orElse(-1)
        );
    }
}
```

OUTPUT



```
/home/aayushi/Downloads/idea-IC-193.6494.35/jbr/bin/java -javaager
4

Process finished with exit code 0
```