

- 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

Sol

```
5
6 package Java8Day1.Ques1;
7
8 interface greater {
9     void test(int var1, int var2);
10}
11
```

```
5
6 package Java8Day1.Ques1;
7
8 public class GreaterThan {
9     public GreaterThan() {
10    }
11
12     public static void main(String[] args) {
13         greater res = (a, b) -> {
14             if (a > b) {
15                 System.out.println("true");
16             } else {
17                 System.out.println("false");
18             }
19         };
20         res.test(12, 15);
21     }
22 }
23
```

```
GreaterThan x
/home/anand/.sdkman/candidates/java/8.0.242-zulu/bin/java ...
false

Process finished with exit code 0
```

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

Sol

```
5
6  package Java8Day1.Ques1;
7
8  interface increment {
9      void test(int var1);
10 }
11
```

```
5
6  package Java8Day1.Ques1;
7
8  public class Increment {
9      public increment() {
10      }
11
12      public static void main(String[] args) {
13          increment res = (a) -> {
14              +a;
15              System.out.println(a);
16          };
17          res.test( 12 );
18      }
19
20 }
```

```
Increment x
/home/aman/.sdkman/candidates/java/8.0.242-zulu/bin/java ...
13
Process finished with exit code 0

  TODO  Terminal
```

- o (3) Concatenation of 2 string Parameter (String , String) Return (String)

Sol

```
5  
6 package Java8Day1.Ques1;  
7  
8 cl interface concatenating {  
9   void test(String var1, String var2);  
10 }  
11
```

```
5  
6 package Java8Day1.Ques1;  
7  
8 ► public class Concatenate {  
9   ►   public Concatenate() {  
10     ►   }  
11  
12 ►   public static void main(String[] args) {  
13     ►   concatenating res = (a, b) -> {  
14     ►     System.out.println(a + b);  
15   ►   };  
16   ►   res.test( a: "Company ", b: "TTN");  
17 ► }  
18 }  
19
```

```
Concenate x  
/home/aman/.sdkman/candidates/java/8.0.242-zulu/bin/java ...  
Company TTN  
Process finished with exit code 0
```

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

Sol

```
5
6     package Java80Day1.Ques1;
7
8     interface toUpper {
9         void test(String var1);
10    }
11
```

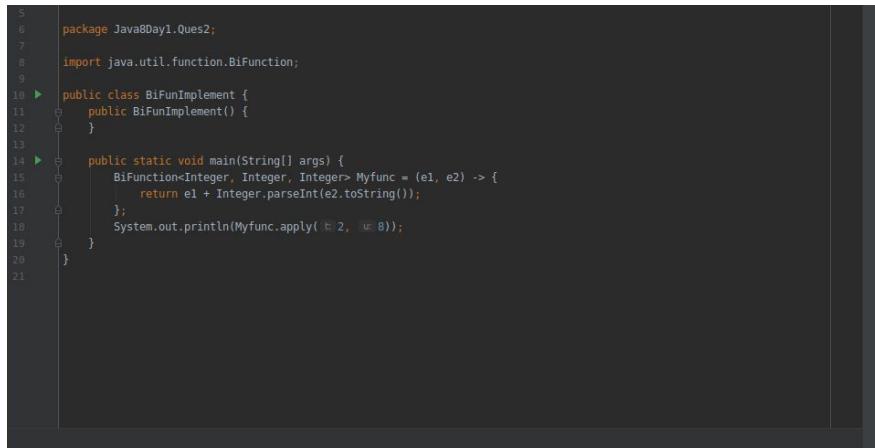
```
6     package Java80Day1.Ques1;
7
8     public class Upper {
9         public Upper() {
10        }
11
12     public static void main(String[] args) {
13         toUpper res = (a) -> {
14             System.out.println(a.toUpperCase());
15         };
16         res.test("company");
17     }
18 }
```

```
Upper ×
/home/aman/.sdkman/candidates/java/8.0.242-zulu/bin/java ...
COMPANY

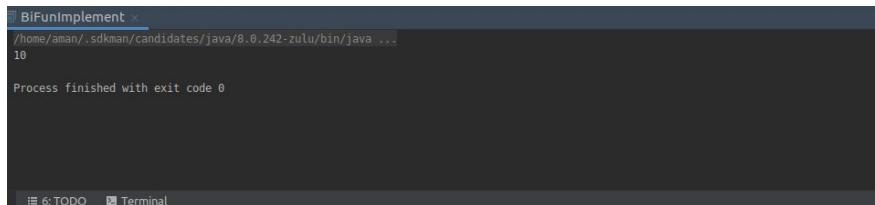
Process finished with exit code 0
```

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

Sol



```
5 package Java8Day1.Ques2;
6
7 import java.util.function.BiFunction;
8
9
10 public class BiFunImplement {
11     public BiFunImplement() {
12     }
13
14     public static void main(String[] args) {
15         BiFunction<Integer, Integer, Integer> Myfunc = (e1, e2) -> {
16             return e1 + Integer.parseInt(e2.toString());
17         };
18         System.out.println(Myfunc.apply( 2,  8));
19     }
20 }
21
```



```
BiFunImplement ×
/home/aman/.sdkman/candidates/java/8.0.242-zulu/bin/java ...
10

Process finished with exit code 0
```

The terminal output shows the command `java` run from the directory `/home/aman/.sdkman/candidates/java/8.0.242-zulu/bin/`. The process ID is 10. The message "Process finished with exit code 0" is displayed at the end.

- 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.

Sol

```
5 package Java8Day1.Ques3;
6
7 interface forInstance1 {
8     void add(int var1, int var2);
9 }
10
11
```

```
6 package Java8Day1.Ques3;
7
8 interface forInstance2 {
9     void sub(int var1, int var2);
10 }
11
```

```
6 package Java8Day1.Ques3;
7
8 interface forStatic {
9     void say(int var1);
10 }
11
```

```
5 package Java8Day1.Ques3;
6
7
8 public class MethodReference {
9     public MethodReference() {
10     }
11
12     public static void saySomething(int i) { System.out.println("Multiplication is:" + i * i); }
13
14     public void saySomething2(int i, int j) { System.out.println("Addition is:" + (i + j)); }
15
16     public void saySomething3(int i, int j) { System.out.println("Subtraction is:" + (i - j)); }
17
18     public static void main(String[] args) {
19         forStatic says = MethodReference::saySomething;
20         System.out.println("Using static method reference:");
21         says.say(8);
22         MethodReference methodReference = new MethodReference();
23         forInstance1 says2 = methodReference::saySomething2;
24         System.out.println("\nUsing instance method reference:");
25         says2.add(10, 20);
26         MethodReference var10000 = new MethodReference();
27         forInstance2 sayable2 = var10000::saySomething3;
28         sayable2.sub(15, 8);
29     }
30 }
```

```
MethodReference <
/home/aman/.sdkman/candidates/java/8.0.242-zulu/bin/java ...
Using static method reference:
Multiplication is:64

Using instance method reference:
Addition is:30
Subtraction is:7

Process finished with exit code 0
```

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

Sol

```

5
6 package Java8Day1.Ques4;
7
8 interface detail {
9     Employee get(String var1, String var2, Integer var3);
10 }
11

```

```

5
6 package Java8Day1.Ques4;
7
8 class Employee {
9     String name;
10    String city;
11    Integer age;
12
13    public Employee(String name, String city, Integer age) {
14        this.name = name;
15        this.city = city;
16        this.age = age;
17    }
18
19    public String toString() { return "name: " + this.name + " city: " + this.city + " age: " + this.age; }
20
21
22
23

```

```

6 package Java8Day1.Ques4;
7
8 public class ConstructorReference {
9     public ConstructorReference() {
10
11
12     public static void main(String[] args) {
13         detail emp = Employee::new;
14         System.out.println(emp.get("Aman", "Gurgaon", 22));
15     }
16
17

```

```

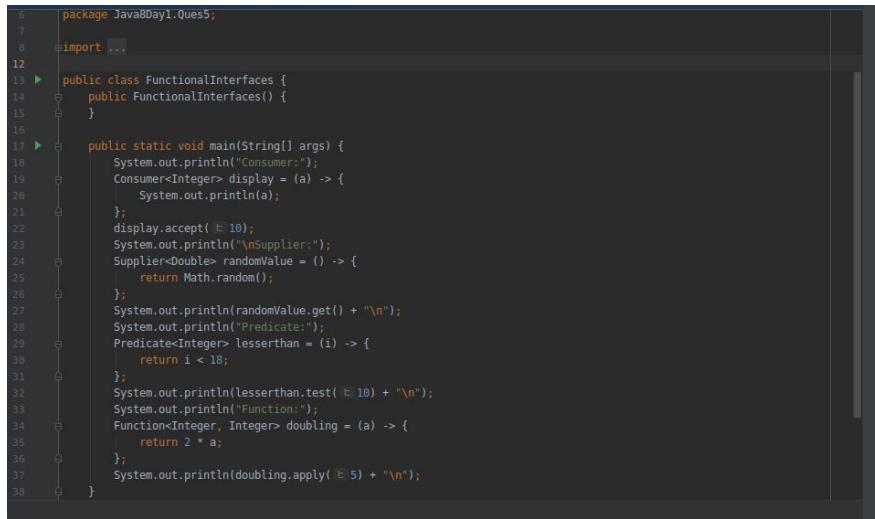
ConstructorReference x
/home/aman/.sdkman/candidates/java/8.0.242-zulu/bin/java ...
name: Aman city: Gurgaon age: 22

Process finished with exit code 0

```

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

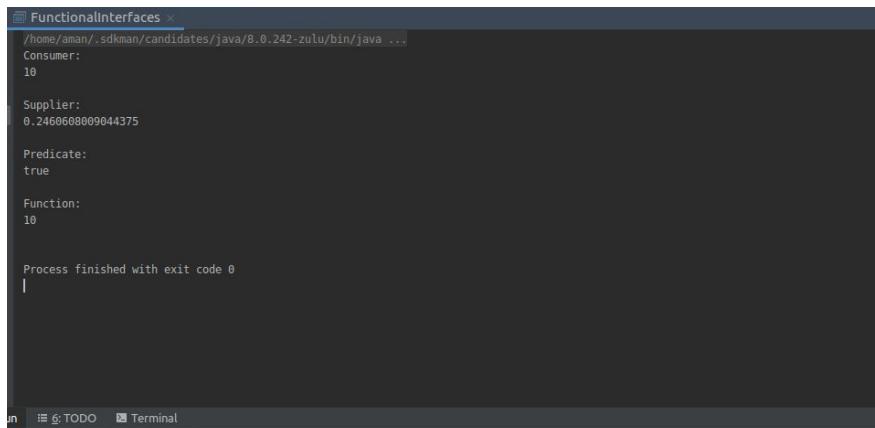
Sol



```

6 package Java8Day1.Ques5;
7
8 import ...
12
13 public class FunctionalInterfaces {
14     public FunctionalInterfaces() {
15     }
16
17     public static void main(String[] args) {
18         System.out.println("Consumer:");
19         Consumer<Integer> display = (a) -> {
20             System.out.println(a);
21         };
22         display.accept(10);
23         System.out.println("\nSupplier:");
24         Supplier<Double> randomValue = () -> {
25             return Math.random();
26         };
27         System.out.println(randomValue.get() + "\n");
28         System.out.println("Predicate:");
29         Predicate<Integer> lesserthan = (i) -> {
30             return i < 18;
31         };
32         System.out.println(lesserthan.test(10) + "\n");
33         System.out.println("Function:");
34         Function<Integer, Integer> doubling = (a) -> {
35             return 2 * a;
36         };
37         System.out.println(doubling.apply(5) + "\n");
38     }
}

```



```

FunctionalInterfaces x
/home/aman/.sdkman/candidates/java/8.0.242-zulu/bin/java ...
Consumer:
10

Supplier:
0.2460608009044375

Predicate:
true

Function:
10

Process finished with exit code 0
|

```

The terminal output shows the execution of the Java code. It prints "Consumer:" followed by the value "10". It then prints "Supplier:" followed by a random double value starting with "0.246...". It prints "Predicate:" followed by "true". Finally, it prints "Function:" followed by the value "10". The process exits with a code of 0.

- Create and access default and static method of an interface.

Sol

```
5 package Java8Day1;
6
7 interface test {
8     static void hello() { System.out.println("Static method of interface."); }
9
10    default void show() { System.out.println("Default method of interface"); }
11
12 }
13
14
15
16
17
```

```
5 package Java8Day1;
6
7
8 public class CreateAndAccess implements test {
9     public CreateAndAccess() {
10
11
12     public static void main(String[] args) {
13         CreateAndAccess obj = new CreateAndAccess();
14         test.hello();
15         obj.show();
16     }
17 }
18
```

```
CreateAndAccess x
/home/aman/.sdkman/candidates/java/8.0.242-zulu/bin/java ...
Static method of interface.
Default method of interface

Process finished with exit code 0
```

- Override the default method of the interface.

Sol

```
5 package Java8Day1;
6
7 interface TestInterface1 {
8     default void show() { System.out.println("Default TestInterface1"); }
9 }
10
11
12
13
```

```
5 package Java8Day1;
6
7
8 public class OverrideDefault implements MyInterface1 {
9     public OverrideDefault() {
10     }
11
12     public void display() { System.out.println("display method of MyInterface1 overwritten."); }
13
14
15 public static void main(String[] args) {
16     OverrideDefault obj = new OverrideDefault();
17     obj.display();
18 }
19
20
21
```

Event Log
21:1 LF UTF-8 4 spaces

```
OverrideDefault x
/home/aman/.sdkman/candidates/java/8.0.242-zulu/bin/java ...
display method of MyInterface1 overwritten.

Process finished with exit code 0
```

- Implement multiple inheritance with default method inside interface.

Sol

```
5
6 package Java8Day1;
7
8 interface TestInterface1 {
9     default void show() { System.out.println("Default TestInterface1"); }
10 }
11
12
13
```

```
5
6 package Java8Day1;
7
8 interface TestInterface2 {
9     default void show() { System.out.println("Default TestInterface2"); }
10 }
11
12
13
```

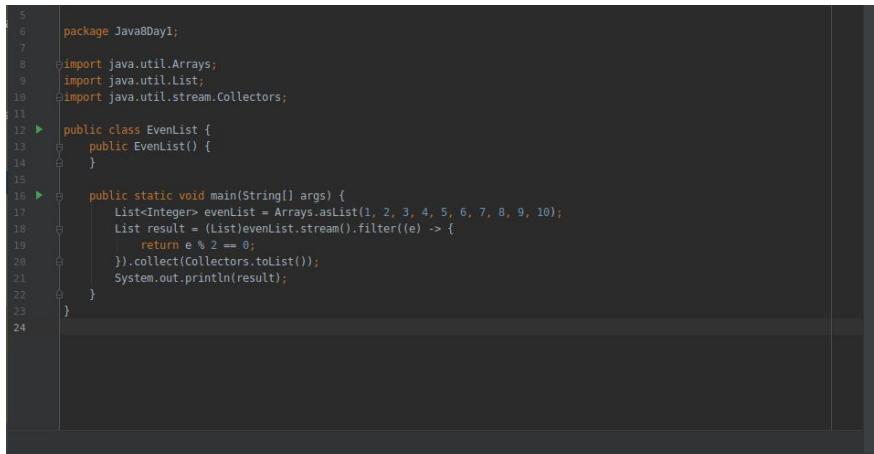
```
6 package Java8Day1;
7
8 class TestClass implements TestInterface1, TestInterface2 {
9     TestClass() {
10     }
11
12     public void show() {
13         super.show();
14         super.show();
15     }
16
17     public static void main(String[] args) {
18         TestClass d = new TestClass();
19         d.show();
20     }
21 }
22
```

```
TestClass x
/home/aman/.sdkman/candidates/java/8.0.242-zulu/bin/java ...
Default TestInterface1
Default TestInterface2

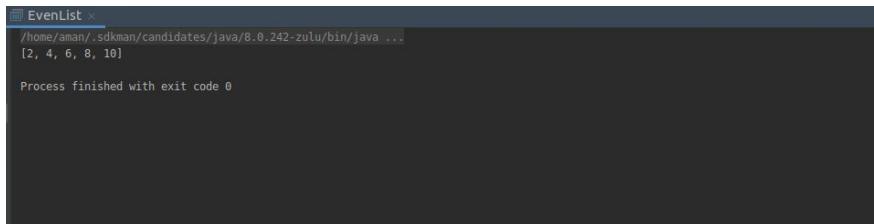
Process finished with exit code 0
```

- Collect all the even numbers from an integer list.

Sol



```
5
6 package Java8Day1;
7
8 import java.util.Arrays;
9 import java.util.List;
10 import java.util.stream.Collectors;
11
12 > public class EvenList {
13     <> public EvenList() {
14     }
15
16 >     public static void main(String[] args) {
17         List<Integer> evenList = Arrays.asList(1, 2, 3, 4, 5, 6, 7, 8, 9, 10);
18         List result = (List)evenList.stream().filter(e) -> {
19             return e % 2 == 0;
20         }.collect(Collectors.toList());
21         System.out.println(result);
22     }
23 }
24
```



```
EvenList ×
/home/aman/.sdkman/candidates/java/8.0.242-zulu/bin/java ...
[2, 4, 6, 8, 10]

Process finished with exit code 0
```

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

Sol

```
5  package Java8Day1;
6
7  import java.util.Arrays;
8  import java.util.List;
9  import java.util.stream.Collectors;
10
11 > public class SumAll {
12     <public SumAll() {
13     }
14
15     <public static void main(String[] args) {
16         List<Integer> evenList = Arrays.asList(1, 2, 3, 4, 5, 6, 7, 8, 9, 10);
17         Integer result = (Integer)evenList.stream().filter((e) -> {
18             <return e > 5;
19         }).collect(Collectors.summingInt((e) -> {
20             <return e;
21         }));
22     <System.out.println(result);
23
24     }
25 }
```

The screenshot shows a terminal window titled "SumAll". The command entered is "/home/aman/.sdkman/candidates/java/8.0.242-zulu/bin/java ...". The output shows the number "40" followed by the message "Process finished with exit code 0". At the bottom of the terminal, there are icons for "TODO" and "Terminal".

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

Sol

```
5
6 package Java8Day1;
7
8 import java.util.Arrays;
9 import java.util.List;
10 import java.util.stream.Collectors;
11
12 > public class AverageAll {
13     public AverageAll() {
14     }
15
16 >     public static void main(String[] args) {
17         List<Integer> evenList = Arrays.asList(1, 2, 3, 4, 5, 6, 7, 8, 9, 10);
18         Double result = (Double)evenList.stream().map((e) -> {
19             return e * 2;
20         }).collect(Collectors.averagingInt((e) -> {
21             return e;
22         }));
23         System.out.println(result);
24     }
25 }
26 }
```

The screenshot shows an IDE interface with two panes. The left pane displays the code for `AverageAll.java`. The right pane shows the terminal output of running the program. The terminal window title is "AverageAll". The command run was `/home/aman/.sdkman/candidates/java/8.0.242-zulu/bin/java ...`. The output shows the result `11.0`, indicating the average of the doubled integers (2, 4, 6, 8, 10).

```
AverageAll x
/home/aman/.sdkman/candidates/java/8.0.242-zulu/bin/java ...
11.0

Process finished with exit code 0
```

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

Sol

```
5  package Java8Day1;
6
7  import java.util.Arrays;
8  import java.util.List;
9  import java.util.Optional;
10
11 > public class FirstEven {
12   >   public FirstEven() {
13   >   }
14
15
16 >   public static void main(String[] args) {
17     >     List<Integer> evenList = Arrays.asList(1, 2, 3, 4, 5, 6, 7, 8, 9, 10);
18     >     Optional<Integer> result = evenList.stream().filter(e -> {
19     >       >       return e % 2 == 0;
20     >     }).filter(e -> {
21     >       >       return e > 3;
22     >     }).findFirst();
23     >     if (result.isPresent()) {
24     >       System.out.println(result);
25     >     } else {
26     >       System.out.println("No result found");
27     >     }
28
29   > }
30
31 }
```

The screenshot shows a terminal window titled "FirstEven". The command entered is "/home/aman/.sdkman/candidates/java/8.0.242-zulu/bin/java ...". The output shows the execution of the code, resulting in the output "Optional[4]". Below the terminal window, there is a dark interface with a "Terminal" tab and other icons.

```
FirstEven x
/home/aman/.sdkman/candidates/java/8.0.242-zulu/bin/java ...
Optional[4]

Process finished with exit code 0

n ☰ TODO Terminal
```