Java 8 features

**JB\_OOP\_LambdaExpression**

MCQ

1. **Correct**

Question: 1

What is a lambda expression in Java 8?

* 1. 

a) A lightweight thread

* 1. 

b) A lightweight process

* 1. Correct Answer



c) A concise way to represent an anonymous function

* 1. 

d) A type of loop construct

Answered

1. **Correct**

Question: 2

What is the syntax for a lambda expression in Java?

* 1. 

a) (parameters) -> expression

* 1. 

b) (parameters) -> { statements }

* 1. 

c) (parameters) -> { return expression; }

* 1. Correct Answer



d) All of the above

Answered

1. **Incorrect**

Question: 3

What is the target type for a lambda expression?

* 1. 

a) A class

* 1. Correct Answer



b) A functional interface

* 1. 

c) A variable

* 1. Incorrect Answer



d) A method Explanation

Answered

1. **Incorrect**

Question: 4

What is the purpose of the -> operator in a lambda expression?

* 1. Correct Answer



a) It separates the parameter list from the body of the lambda expression

* 1. 

b) It specifies the return type of the lambda expression

* 1. 

c) It denotes the start and end of the lambda expression

* 1. Incorrect Answer



d) It represents the “arrow” indicating a lambda expression

Answered

1. **Correct**

Question: 5

What will the following lambda expression return?  
(x, y) -> x + y

* 1. Correct Answer



A) Sum of x and y

* 1. 

B) Product of x and y

* 1. 

C) Difference of x and y

* 1. 

D) Quotient of x and y

Answered

1. **Correct**

Question: 6

Lambda expressions can be used to replace …

* 1. 

A) Objects

* 1. 

B) Variables

* 1. Correct Answer



C) Anonymous classes

* 1. 

D) Loops

Answered

1. **Correct**

Question: 7

Which of the following is a valid lambda expression?

* 1. 

A) (int x, int y) -> x + y

* 1. 

B) (x, y) -> return x + y;

* 1. 

C) x -> x\*x

* 1. Correct Answer



D) All of the above

Answered

1. **Correct**

Question: 8

Lambda expressions can be used with the Java Collections API primarily in **\_**

* 1. 

A) Sorting

* 1. 

B) Iterating

* 1. 

C) Filtering

* 1. Correct Answer



D) All of the above

Answered

1. **Incorrect**

Question: 9

The scope of variables in a lambda expression is …

* 1. 

A) Global

* 1. Incorrect Answer



B) Local to the lambda expression

* 1. Correct Answer



C) The same as the enclosing scope

* 1. 

D) None

Answered

1. **Correct**

Question: 10

You cannot use … inside lambda expressions.

* 1. 

A) return statement

* 1. 

B) this keyword

* 1. Correct Answer



C) break and continue

* 1. 

D) arithmetic operations

Answered

1. **Incorrect**

Question: 11

Which Java feature works effectively with lambda expressions?

* 1. Incorrect Answer



A) Generics

* 1. 

B) Enums

* 1. 

C) Polymorphism

* 1. Correct Answer



D) Streams

Answered

1. **Correct**

Question: 12

Lambda expressions can throw exceptions?

* 1. Correct Answer



A) True

* 1. 

B) False

Answered

1. **Correct**

Question: 13

What does the following lambda expression represent?  
() -> {}

* 1. 

A) A lambda that accepts two parameters and does nothing

* 1. Correct Answer



B) A lambda that does nothing and returns void

* 1. 

C) A lambda that throws an exception

* 1. 

D) None of the above

Answered

1. **Correct**

Question: 14

How is this keyword handled inside a lambda expression?

* 1. 

A). You can’t use this inside a lambda expression

* 1. 

B). this refers to the functional interface of the lambda expression

* 1. 

C). this refers to the lambda expression itself

* 1. Correct Answer



D). this refers to the enclosing class of the lambda expression

Answered

1. **Correct**

Question: 15

What is the benefit of using lambda expressions in Java?

* 1. 

a) Improved performance

* 1. Correct Answer



b) More readable and concise code

* 1. 

c) Enhanced error handling

* 1. 

d) Higher memory utilization

Answered

1. **Correct**

Question: 16

What does SAM stand for in the context of Functional Interface?

* 1. 

a) Single Ambivalue Method

* 1. Correct Answer



b) Single Abstract Method

* 1. 

c) Simple Active Markup

* 1. 

d) Simple Abstract Markup

Answered

[**Submit**](https://codehs.com/student/4767136/section/512694/assignment/133574248)

Display a message using Lambda expression

Q1.Using Functional Interface with Lambda Expression

WAP for the below requirement:

  i.Create a functional interface with method: displayMessage()

  ii.Use lambda expression to implement functional interface

  iii.Call the method using lambda expression

@FunctionalInterface

**interface** Functional\_Interface

{

**void** displayMessage();

}

//---------------------Main class-----------------------------------

**public** **class** MyProgram// Main

{

**public** **static** **void** main(String[] args)

{

Functional\_Interface fi=()-> System.***out***.println("Lambda expression");

fi.displayMessage();

}

}

Calculator program using Lambda

WAP for the below requirement:

  i. Create a functional Interface calculator with method calculate()

   Method will have parameters to receive two decimal nos.

  ii. Using lambda expression implement functional interface

    and find the sum,difference,product,quotient,remainder

  iii. Call the methods using lambda expressions

**import** java.util.Scanner;

@FunctionalInterface

**interface** Calculator

{

**void** calculate(**double** a,**double** b);

}

//---------------------Main class-----------------------------------

**public** **class** MyProgram// Main

{

**public** **static** **void** main(String[] args)

{

Scanner sc=**new** Scanner(System.***in***);

System.***out***.println("Enter 1st decmial values :");

**double** a=sc.nextDouble();

System.***out***.println("Enter 2nd decmial values :");

**double** b=sc.nextDouble();

Calculator c= (**double** num1,**double** num2) ->

{

// find the sum,difference,product,quotient,remainder

System.***out***.println("Sum :"+(num1+num2));

System.***out***.println("Difference :"+Math.*abs*(num1-num2));

System.***out***.println("Product :"+(num1\*num2));

System.***out***.println("Quotient :"+(num1/num2));

System.***out***.println("Remainder :"+(num1%num2));

};

c.calculate(a, b);

}

}

**Output:-**

Enter 1st decmial values :

10.0

Enter 2nd decmial values :

2.0

Sum :12.0

Difference :8.0

Product :20.0

Quotient :5.0

Remainder :0.0

Hello World Program

1) WAP to print Hello World Program using Functional Interface and

  Lambda Expression

@FunctionalInterface

**interface** Functional\_Interface

{

**void** display();

}

//---------------------Main class-----------------------------------

**public** **class** MyProgram// Main

{

**public** **static** **void** main(String[] args)

{

Functional\_Interface fi=()-> System.***out***.println("Hello World");

fi.display();

}

}

Reverse a string using Lambda

WAP to Reverse a String using Functional Interface and Lambda

  Expression

@FunctionalInterface

**interface** Functional\_Interface

{

**void** display(String s);

}

//---------------------Main class-----------------------------------

**public** **class** MyProgram// Main

{

**public** **static** **void** main(String[] args)

{

Functional\_Interface fi=(String s)->

{

**int** a=s.length()-1;

**for**(**int** i=a;i>-1;i--)

{

System.***out***.print(s.charAt(i));

}

};

fi.display("Sourab das");

}

}

Calculate Square of the number using Lambda

WAP to calculate the Square of a Number using Lambda Expression

@FunctionalInterface

**interface** Functional\_Interface

{

**void** square(**int** a);

}

//---------------------Main class-----------------------------------

**public** **class** MyProgram// Main

{

**public** **static** **void** main(String[] args)

{

Functional\_Interface fi=(**int** num)->

System.***out***.println("Square of a Number "+num+" is :"+(num\*num));

fi.square(25);

}

}

Palindrome program using Lambda

WAP to Check if a String is Palindrome using Lambda Expression

**import** java.util.Scanner;

@FunctionalInterface

**interface** Functional\_Interface

{

**void** palindrome(String s);

}

//---------------------Main class-----------------------------------

**public** **class** MyProgram// Main

{

**public** **static** **void** main(String[] args)

{

Scanner sc=**new** Scanner(System.***in***);

System.***out***.print("Enter a String :");

String st=sc.nextLine();

Functional\_Interface fi=(String s)->

{

**int** a=s.length()-1;

**int** j=0,f=0;

**for**(**int** i=a;i>a/2;i--)

{

**if**(s.charAt(i)==s.charAt(j))

{

j++;

f++;

}

**else**

{

f=0;

**break**;

}

}

**if**(f>0)

System.***out***.println("palindrome");

**else** System.***out***.println("Not a palindrome");

};

fi.palindrome(st);

}

}

Find Max value of an Array using Lambda

WAP to find the Maximum Element in an Array using Lambda Expression

**import** java.util.Scanner;

@FunctionalInterface

**interface** Functional\_Interface

{

**void** Max(**int** [] a);

}

//---------------------Main class-----------------------------------

**public** **class** MyProgram// Main

{

**public** **static** **void** main(String[] args)

{

System.***out***.println("Enter size of an array :");

Scanner sc=**new** Scanner(System.***in***);

**int** size=sc.nextInt();

**int** []arr=**new** **int** [size] ;

System.***out***.println("Enter "+size+" elements");

**for**(**int** i=0;i<size;i++)

{

arr[i]=sc.nextInt();

}

Functional\_Interface fi=(**int** [] a)->

{

**int** i=0,max=a[i];

**for**(i=1;i<a.length;i++)

{

**if**(a[i]>max)max=a[i];

}

System.***out***.println("Maximun no is :"+max);

};

fi.Max(arr);

}

}

MCQ1

1. **Incorrect**

Question: 1

@FunctionalInterface  
interface Invisible {  
public void show();  
}

class Display implements Invisible {  
Invisible inv;  
public void show() {  
inv = () -> System.out.println(“I am Visible”);  
;  
}  
}

public class TestMcq {  
public static void main(String[] args) {  
new Display().inv.show();  
}  
}

* 1. 

A. Compile time error

* 1. Correct Answer



B. Runtime Exception(NullpointerException)

* 1. Incorrect Answer



C. I am Visible

* 1. 

D. I am Visible I am Visible

* 1. 

E. No Output

Answered

1. **Incorrect**

Question: 2

interface Invisible {  
public void show();  
}

class Display implements Invisible {  
Invisible inv;  
public void show() {  
inv = () -> System.out.println(“I am Visible”);

}

}

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

Display inv = new Display();

inv.show();

inv.inv.show();

}

}

* 1. 

A. Compile time error

* 1. Incorrect Answer



B. Runtime Exception(NullpointerException)

* 1. Correct Answer



C. I am Visible

* 1. 

D. I am Visible I am Visible

* 1. 

E. No Output

Answered

1. **Correct**

Question: 3

@FunctionalInterface  
interface Readable {  
public String read(String name);  
}  
class Book implements Readable {  
Readable r = (name) ->{  
return name;  
};  
@Override  
public String read(String name) {  
return name +” “+ r.read(name);  
}

}  
public class TestMcq {  
public static void main(String[] args) {  
Book b = new Book();  
System.out.println(b.read(“Elevation”));  
System.out.println(b.r);

}

}

* 1. 

A. Compile time error

* 1. 

B. Runtime Error

* 1. 

C. Elevation  
Elevation

* 1. 

D. Elevation Elevation  
Elevation

* 1. Correct Answer



E. Elevation Elevation  
lambda.Book$$Lambda$1/0x00800c00df0@2a84aee7

Answered

1. **Correct**

Question: 4

@FunctionalInterface  
interface Readable {  
String name = “The Stranger”;

public Readable read();

}

class Book implements Readable {  
Readable r;

@Override

public Readable read() {

r = () -> {

System.out.println(r.name);

return r;

};

return r;

}

}

public class TestMcq {  
public static void main(String[] args) {  
System.out.println(new Book().read().read().name);

}

}

* 1. 

A. Compile time error

* 1. 

B. Runtime Error

* 1. Correct Answer



C. The Stranger  
The Stranger

* 1. 

D. The Stranger

* 1. 

E. The Stranger  
The Stranger  
The Stranger  
The Stranger

Answered

1. **Correct**

Question: 5

@FunctionalInterface  
interface Readable {  
String name = “The Stranger”;

public Readable read();

}

class Book implements Readable {  
Readable r = read();

@Override

public Readable read() {

r = () -> {

System.out.println(Book.name);

return this;

};

return r;

}

}

public class TestMcq {  
public static void main(String[] args) {  
Book b1 = new Book().read();  
b1.read();  
}  
}

* 1. Correct Answer



A. Compile time error

* 1. 

B. Runtime Error

* 1. 

C. The Stranger  
The Stranger

* 1. 

D. The Stranger

* 1. 

E. The Stranger  
The Stranger  
The Stranger

Answered

1. **Correct**

Question: 6

@FunctionalInterface  
interface Adorable {  
int level = 10;  
void smile();  
}  
@FunctionalInterface  
interface Lovable extends Adorable{  
int level = 20;  
}  
public class TestMcq {  
public static void main(String[] args) {  
Adorable a;  
Adorable adob = new Lovable() {  
@Override  
public void smile() {  
a = () -> System.out.println(“Smile you are cute.”);  
}  
};  
a.smile();  
}  
}

* 1. Correct Answer



A. Compile time error

* 1. 

B. Runtime error(NullPointerException)

* 1. 

C. Smile you are cute.

* 1. 

D. No output

Answered

1. **Correct**

Question: 7

interface Sample {  
int x = 20;  
void m1();  
}

class Example {  
Sample s;

public Example(Sample s) {

super();

this.s = s;

}

}

public class TestMCQ {  
public static void main(String[] args) {  
Sample s = () -> System.out.println(Sample.x);  
Example e = new Example(s);  
e.s = null;  
System.out.println(e.s.x);  
s.m1();  
}  
}

* 1. 

A. Compiletime Error

* 1. 

B. Exception at runtime(NullPointerException)

* 1. 

C. 20  
Exception at runtime(NullPointerException)

* 1. 

D. Exception at runtime(NullPointerException)  
20

* 1. Correct Answer



E. 20 20

Answered

1. **Correct**

Question: 8

interface Sample {  
int x = 20;  
int m1();  
}

public class TestMCQ {  
public static void main(String[] args) {  
Sample s = new Sample() {  
public int m1() {  
x +=10;  
return x;  
}  
};  
System.out.println(s.m1());  
}  
}

* 1. Correct Answer



A. Compiletime Error

* 1. 

B. 20

* 1. 

C. 30

* 1. 

D. No output

Answered

1. **Correct**

Question: 9

interface Sample {  
int x = 20;

public static void m1() {

System.out.println("m1 from interface");

}

}

class Example {  
Sample s;

public Example(Sample s) {

super();

this.s = s;

}

}

public class TestMcq {  
public static void main(String[] args) {  
Sample s = new Sample() {  
int x = 30;

public static void m1() {

System.out.println("m1 from AIC");

}

};

Example e = new Example(s);

System.out.println(e.s.x);

e.s.m1();

}

}

* 1. Correct Answer



A. Compiletime error

* 1. 

B. 30  
m1 from AIC

* 1. 

C. 20  
m1 from interface

* 1. 

D. 20  
m1 from AIC

* 1. 

E. 30  
m1 from interface

Answered

1. **Correct**

Question: 10

interface Sample {  
int x = 20;

void m1();

}

class Example implements Sample {  
void m1() {  
System.out.println(“Overriden method”);  
}

void m2() {

System.out.println("m2 executed");

}

}

public class TestMCQ {  
public static void main(String[] args) {  
Example e = new Example() {  
void m2() {  
m1();  
}  
};  
e.m2();  
}  
}

* 1. Correct Answer



A.Compiletime error

* 1. 

B.Overriden method

* 1. 

C.Overriden method  
m2 executed

* 1. 

D.m2 executed  
Overriden method

Answered

1. **Correct**

Question: 11

interface Sample {  
int x = 20;  
void add(int x, int y);  
}

class Example {  
Sample s;

public Example(Sample s) {

super();

this.s = s;

}

}

public class TestMCQ {  
public static void main(String[] args) {  
Sample s = (x,y)->System.out.println(x+y);  
Example e = new Example(s);  
e.s = null;  
System.out.println(s.x);  
e.s.add(10,20);  
}  
}

* 1. 

A. Exception in thread “main” java.lang.NullPointerException

* 1. 

B. 20 30

* 1. 

C. 20

* 1. 

D. Compile time Error

* 1. Correct Answer



E. 20  
Exception in thread “main” java.lang.NullPointerException

Answered

1. **Correct**

Question: 12

interface I1 {  
int x;  
private void m1() {  
x = 10;  
System.out.println(x);  
}  
default void m2() {  
System.out.println(“m2 from I1”);  
m1();  
}  
}  
class A implements I1{  
public void m2() {  
I1.super.m2();  
}  
}  
public class TestMcq {  
public static void main(String[] args) {  
A a = new A();  
a.m2();  
}  
}

* 1. Correct Answer



A. Compiletime Error

* 1. 

B. 10

* 1. 

C. LinkageError

* 1. 

D. m2 from I1  
10

Answered

1. **Correct**

Question: 13

interface I  
{  
int myInterface(int a, int b);  
}  
public class TestMcq {  
public static void main(String[] args) {  
I i = (a, b) ->  
{  
int div = a/b;  
int addition = a+b;  
return addition;  
};  
}  
}

* 1. 

1

* 1. 

2

* 1. Correct Answer



no output

* 1. 

Compile time error

* 1. 

Runtime Exception

Answered

1. **Correct**

Question: 14

interface NIT  
{  
int m1(int a, int b);  
}  
public class TestMcq {  
public static void main(String[] args) {  
NIT nit = (int a, int b) -> System.out.println(a+b);  
nit.m1(5, 10);  
}  
}

* 1. Correct Answer



Compile time error

* 1. 

15

* 1. 

510

* 1. 

Runtime exception

Answered

1. **Correct**

Question: 15

interface NIT {  
String meth();  
}

public class TestMcq {  
public static void main(String[] args) {  
NIT nit = () -> “NareshITechnologies”;  
nit.meth();  
}  
}

* 1. 

NareshITechnologies

* 1. 

Compile time error

* 1. Correct Answer



No Output

* 1. 

Runtime exception

Answered

1. **Correct**

Question: 16

interface Movie {  
String meth();  
}

public class TestMcq {  
public static void main(String[] args) {  
Movie m = () -> return “Salar”;  
System.out.println(m.meth());  
}  
}

* 1. 

Salar

* 1. Correct Answer



Compile time error

* 1. 

No output

* 1. 

Runtime Exception

Answered

1. **Correct**

Question: 17

What is the output of the following code  
@FunctionalInterface  
interface NIT  
{  
String nit(String name);  
}  
public class FunctionalInterfaceExample  
{  
public static void main(String[] args) {  
NIT nit = (String name) -> “Welcome to “;  
System.out.println(nit.nit(“NARESHIT”));  
}  
}

* 1. Correct Answer



Welcome to

* 1. 

Welcome to NARESHIT

* 1. 

Welcome toNARESHIT

* 1. 

No outpit

Answered

1. **Incorrect**

Question: 18

What is the output of the following code  
@FunctionalInterface  
interface NIT  
{  
int cal(int value);  
}  
public class FunctionalInterfaceExample  
{  
public static void main(String[] args) {  
NIT nit = (int value) -> value+75;  
System.out.println(“Result: “+nit.cal(25));  
}  
}

* 1. Incorrect Answer



100

* 1. 

25

* 1. 

75

* 1. Correct Answer



Result 100

Answered

1. **Correct**

Question: 19

What is the output of the following Snippet  
public class FunctionalInterface {  
public static void main(String args[])  
{  
Function div = a -> a / 2;  
System.out.println(div.apply(15));  
}  
}

* 1. 

15

* 1. 

7

* 1. 

7.5

* 1. Correct Answer



Compile time error

Answered

1. **Correct**

Question: 20

interface NIT  
{  
void myInterface(int a);  
default void defalutFunction()  
{  
System.out.println(“Naresh I Technologies”);  
}

}  
public class Java8 {  
public static void main(String[] args) {  
NIT nit = (int a) -> System.out.println(a/5);  
nit.myInterface(15);  
}  
}

* 1. Correct Answer



3

* 1. 

7

* 1. 

3.5

* 1. 

7.5

* 1. 

Compile time error

Answered

1. **Correct**

Question: 21

@FunctionalInterface  
interface NIT  
{  
String nit(String courseName);  
}  
interface Course extends NIT  
{  
String courseName();  
String facultyName();  
String duration();  
}  
public class FunctionalInterfaceExample  
{  
public static void main(String[] args) {  
NIT nit = (String course) -> { return course; };  
nit = null;  
System.out.println(nit.nit(“Java”));  
}  
}

* 1. 

Compile time error

* 1. Correct Answer



NullPointerException

* 1. 

Java

* 1. 

No output

Answered

1. **Incorrect**

Question: 22

Choose the correct output of the following program  
interface Drawable  
{  
void draw();  
}  
public class FunctionalInterface {

public static void main(String[] args)

{

() -> System.out.println("Drawing...");

}

}

* 1. Correct Answer



Compile time error

* 1. 

Drawing…

* 1. Incorrect Answer



Runtime exception

* 1. 

No output

Answered

1. **Correct**

Question: 23

What is the output of the following code?

interface NIT  
{  
public int myInterface(int a);

}  
public class Java8 {  
public static void main(String[] args) {  
NIT nit = (String a) -> a\*5;  
int i = nit.myInterface(6);  
System.out.println(i);  
}  
}

* 1. 

6

* 1. 

5

* 1. 

30

* 1. Correct Answer



Compile time error

Answered

1. **Correct**

Question: 24

Is there any error in following code?  
@FunctionalInterface  
interface NIT  
{  
String myInterface(String a);  
int hashCode();  
String toString();  
}

* 1. 

yes

* 1. Correct Answer



no

Answered

1. **Incorrect**

Question: 25

What is the output of the following code  
public class FunctionalInterfaceExample{  
public static void main(String[] args) {  
Predicate grater = a -> a >15;

Predicate<Integer> less = a -> a <10;

boolean bool = grater.test(20);

boolean bool1 = less.test(5);

System.out.println(bool);

System.out.println(bool1);

}

}

* 1. 

false  
false

* 1. Correct Answer



true  
true

* 1. Incorrect Answer



true  
false

* 1. 

false  
true

Answered

1. **Correct**

Question: 26

Which is the mandatory condition to define a functional interface in Java 8?

* 1. 

(a) The interface should have @functionalInterface annotation on top of it

* 1. 

(b) The interface should have only one method

* 1. Correct Answer



(c) The interface should have only one abstract method

* 1. 

(d) The interface should at least have one method

Answered

1. **Incorrect**

Question: 27

Which of the following is a valid lambda expression in Java 8?

* 1. Correct Answer



(a) () -> { System.out.println(“Hello”); }

* 1. 

(b) (int x) -> { x++; }

* 1. 

(c) (String s, int x) -> { System.out.println(s + x); }

* 1. Incorrect Answer



(d) All of the above

Answered

1. **Incorrect**

Question: 28

What is the purpose of the Supplier interface in Java 8?

* 1. Correct Answer



(a) To represent a method that takes no arguments and returns no value

(c) To represent a method that takes no arguments and returns a value

* 1. 

(b) To represent a method that takes one argument and returns a value

* 1. Incorrect Answer



(c) To represent a method that takes no arguments and returns a value

* 1. 

(d) To represent a method that takes one argument and returns no value

Answered

1. **Incorrect**

Question: 29

public class TestMcq {  
public static void main(String[] args) {  
Predicate notNull = (arg -> (arg == null));  
System.out.println(notNull.test(null));  
}  
}

* 1. Correct Answer



true

* 1. Incorrect Answer



false

Answered

1. **Correct**

Question: 30

Which one of the following abstract methods does not take any argument but returns a value?

* 1. 

A. The accept() method in java.util.function.Consumer interface

* 1. Correct Answer



B. The get() method in java.util.function.Supplier interface

* 1. 

C. The test() method in java.util.function.Predicate interface

* 1. 

D. The apply() method in java.util.function.Function interface

Answered

[**Submit**](https://codehs.com/student/4767136/section/512694/assignment/131906922)