	codewitharrays.in freelance project available to buy contact on 8007592194	
SR.NO	Project NAME	Technology
1	Online E-Learning Platform Hub	React+Springboot+MySql
2	PG Mates / RoomSharing / Flat Mates	React+Springboot+MySql
3	Tour and Travel management System	React+Springboot+MySql
4	Election commition of India (online Voting System)	React+Springboot+MySql
5	HomeRental Booking System	React+Springboot+MySql
6	Event Management System	React+Springboot+MySql
7	Hotel Management System	React+Springboot+MySql
8	Agriculture web Project	React+Springboot+MySql
9	AirLine Reservation System / Flight booking System	React+Springboot+MySql
10	E-commerce web Project	React+Springboot+MySql
11	Hospital Management System	React+Springboot+MySql
12	E-RTO Driving licence portal	React+Springboot+MySql
13	Transpotation Services portal	React+Springboot+MySql
14	Courier Services Portal / Courier Management System	React+Springboot+MySql
15	Online Food Delivery Portal	React+Springboot+MySql
16	Muncipal Corporation Management	React+Springboot+MySql
17	Gym Management System	React+Springboot+MySql
18	Bike/Car ental System Portal	React+Springboot+MySql
19	CharityDonation web project	React+Springboot+MySql
20	Movie Booking System	React+Springboot+MySql

	freelance_Project available to buy contact on 8007592194	
21	Job Portal web project	React+Springboot+MySql
22	LIC Insurance Portal	React+Springboot+MySql
23	Employee Management System	React+Springboot+MySql
24	Payroll Management System	React+Springboot+MySql
25	RealEstate Property Project	React+Springboot+MySql
26	Marriage Hall Booking Project	React+Springboot+MySql
27	Online Student Management portal	React+Springboot+MySql
28	Resturant management System	React+Springboot+MySql
29	Solar Management Project	React+Springboot+MySql
30	OneStepService LinkLabourContractor	React+Springboot+MySql
31	Vehical Service Center Portal	React+Springboot+MySql
32	E-wallet Banking Project	React+Springboot+MySql
33	Blogg Application Project	React+Springboot+MySql
34	Car Parking booking Project	React+Springboot+MySql
35	OLA Cab Booking Portal	React+NextJs+Springboot+MySql
36	Society management Portal	React+Springboot+MySql
37	E-College Portal	React+Springboot+MySql
38	FoodWaste Management Donate System	React+Springboot+MySql
39	Sports Ground Booking	React+Springboot+MySql
40	BloodBank mangement System	React+Springboot+MySql

41	Bus Tickit Booking Project	React+Springboot+MySql
42	Fruite Delivery Project	React+Springboot+MySql
43	Woodworks Bed Shop	React+Springboot+MySql
44	Online Dairy Product sell Project	React+Springboot+MySql
45	Online E-Pharma medicine sell Project	React+Springboot+MySql
46	FarmerMarketplace Web Project	React+Springboot+MySql
47	Online Cloth Store Project	React+Springboot+MySql
48	Train Ticket Booking Project	React+Springboot+MySql
49	Quizz Application Project	JSP+Springboot+MySql
50	Hotel Room Booking Project	React+Springboot+MySql

51	Online Crime Reporting Portal Project	React+Springboot+MySql
52	Online Child Adoption Portal Project	React+Springboot+MySql
53	online Pizza Delivery System Project	React+Springboot+MySql
54	Online Social Complaint Portal Project	React+Springboot+MySql
55	Electric Vehical management system Project	React+Springboot+MySql
56	Online mess / Tiffin management System Project	React+Springboot+MySql
57		React+Springboot+MySql
58		React+Springboot+MySql
59		React+Springboot+MySql
60		React+Springboot+MySql

Spring Boot + React JS + MySQL Project List

Sr.No	Project Name	YouTube Link
1	Online E-Learning Hub Platform Project	https://youtu.be/KMjyBaWmgzg?si=YckHuNzs7eC84-IW
2	PG Mate / Room sharing/Flat sharing	https://youtu.be/4P9cIHg3wvk?si=4uEsi0962CG6Xodp
3	Tour and Travel System Project Version 1.0	https://youtu.be/-UHOBywHaP8?si=KHHfE_A0uv725f12
4	Marriage Hall Booking	https://youtu.be/VXz0kZQi5to?si=llOS-QG3TpAFP5k7
5	Ecommerce Shopping project	https://youtu.be/vJ_C6LkhrZ0?si=YhcBylSErvdn7paq
6	Bike Rental System Project	https://youtu.be/FIzsAmIBCbk?si=7ujQTJqEgkQ8ju2H
7	Multi-Restaurant management system	https://youtu.be/pvV-pM2Jf3s?si=PgvnT-yFc8ktrDxB
8	Hospital management system Project	https://youtu.be/IynIouBZvY4?si=CXzQs3BsRkjKhZCw
9	Municipal Corporation system Project	https://youtu.be/cVMx9NVyI4I?si=qX0oQt-GT-LR_5jF
10	Tour and Travel System Project version 2.0	https://youtu.be/_4u0mB9mHXE?si=gDiAhKBowi2gNUKZ

Sr.No	Project Name	YouTube Link
11	Tour and Travel System Project version 3.0	https://youtu.be/Dm7nOdpasWg?si=P_Lh2gcOFhlyudug
12	Gym Management system Project	https://youtu.be/J8_7Zrkg7ag?si=LcxV51ynfUB7OptX
13	Online Driving License system Project	https://youtu.be/3yRzsMs8TLE?si=JRI_z4FDx4Gmt7fn
14	Online Flight Booking system Project	https://youtu.be/m755rOwdk8U?si=HURvAY2VnizIyJlh
15	Employee management system project	https://youtu.be/lD1iE3W_GRw?si=Y_jv1xV_BljhrD0H
16	Online student school or college portal	https://youtu.be/4A25aEKfei0?si=RoVgZtxMk9TPdQvD
17	Online movie booking system project	https://youtu.be/Lfjv_U74SC4?si=fiDvrhhrjb4KSlSm
18	Online Pizza Delivery system project	https://youtu.be/Tp3izreZ458?si=8eWAOzA8SVdNwlyM
19	Online Crime Reporting system Project	https://youtu.be/0UlzReSk9tQ?si=6vN0e70TVY1GOwPO
20	Online Children Adoption Project	https://youtu.be/3T5HC2HKyT4?si=bntP78niYH802I7N

1. What does the following Java code snippet output?

```
List<String> list = new ArrayList<>();
list.add("Java");
list.add("Python");
list.add("C++");
System.out.println(list.get(1));
```

a) Java

b) Python

c) C++

d) An IndexOutOfBoundsException is thrown

Click to View Answer and Explanation

Answer:

b) Python

Explanation:

`list.get(1)` retrieves the second element in the list, which is "Python".

2. What is the result of executing this Java code snippet?

```
Set<Integer> set = new HashSet<>();
set.add(1);
set.add(2);
set.add(1);
System.out.println(set.size());
```

a) 2

b) 3

c) 4

d) Compilation error

Click to View Answer and Explanation

Answer:

a) 2

Explanation:

A `HashSet` does not allow duplicate elements. Adding 1 twice does not change its size.

3. What will be printed by this Java code?

```
Map<String, Integer> map = new HashMap<>();
map.put("Alice", 30);
map.put("Bob", 25);
map.put("Alice", 35);
System.out.println(map.get("Alice"));
```

a) 25

b) 30

c) 35

d) null

Click to View Answer and Explanation

Answer:

c) 35

Explanation:

The second `put` operation updates the value associated with "Alice" to 35.

4. Identify the output of the following code:

```
List<String> list = Arrays.asList("A", "B", "C", "D");
for (String s : list) {
    System.out.print(s + " ");
}
```

a) A B C D

b) A, B, C, D

c) [A, B, C, D]

d) An UnsupportedOperationException is thrown

Click to View Answer and Explanation

Answer:

a) A B C D

Explanation:

The enhanced `for` loop iterates over each element in the list and prints it.

5. What does this code snippet output?

```
Queue<Integer> queue = new LinkedList<>();
queue.add(1);
queue.add(2);
queue.add(3);
System.out.println(queue.peek());
```

a) 1

b) 2

c) 3

d) null

Click to View Answer and Explanation

Answer:

a) 1

Explanation:

`peek()` retrieves but does not remove the head of the queue, which is 1.

6. What is the result of executing this code?

```
Deque<Integer> deque = new ArrayDeque<>();
deque.offerFirst(1);
deque.offerLast(2);
System.out.println(deque.pollLast());
```

a) 1

b) 2

c) null

d) An exception is thrown

Click to View Answer and Explanation

Answer:

b) 2

Explanation:

`pollLast()` retrieves and removes the last element of the deque, which is 2.

7. What will the following Java code snippet output?

```
Map<String, String> map = new TreeMap<>();
map.put("c", "C");
map.put("b", "B");
map.put("a", "A");
for (String key : map.keySet()) {
    System.out.print(key + " ");
}
```

a) a b c

b) c b a

c) A B C

d) C B A

Click to View Answer and Explanation

Answer:

a) a b c

Explanation:

A `TreeMap` sorts its keys. The keys are iterated in ascending order.

8. What does the following code snippet print?

```
List<String> list = new ArrayList<>(Arrays.asList("A", "B", "C"));
list.remove("B");
System.out.println(list);
```

a) [A, B, C]

b) [A, C]

c) [B, C]

d) An UnsupportedOperationException is thrown

Click to View Answer and Explanation

Answer:

b) [A, C]

Explanation:

The `remove` method removes "B" from the list, leaving "A" and "C".

9. Determine the output of this Java code:

```
List<Integer> list = new ArrayList<>(Arrays.asList(1, 2, 3, 4, 5));
list.removeIf(n -> n % 2 == 0);
System.out.println(list);
```

a) [1, 2, 3, 4, 5]

b) [1, 3, 5]

c) [2, 4]

d) []

Click to View Answer and Explanation

Answer:

b) [1, 3, 5]

Explanation:

`removeIf` removes elements that match the given predicate, which in this case are the even numbers.

10. What is the result of the following code snippet?

```
Set<String> set = new LinkedHashSet<>(Arrays.asList("A", "B", "C"));
set.add("D");
set.add("B");
System.out.println(set);
```

a) [A, B, C, D]

b) [A, C, D, B]

c) [D, A, B, C]

d) [B, A, C, D]

Click to View Answer and Explanation

Answer:

a) [A, B, C, D]

Explanation:

A `LinkedHashSet` maintains insertion order and does not allow duplicates. "B" is not added again.

11. What will this Java code snippet output?

```java
Map<Integer, String> map = new HashMap<>();
map.put(1, "A");
map.put(2, "B");
map.put(3, "C");
map.replace(2, "D");
System.out.println(map);
```

a) {1=A, 2=B, 3=C}

b) {1=A, 2=D, 3=C}

c) {1=A, 3=C}

d) {1=A, 2=B, 3=C, 2=D}

Click to View Answer and Explanation

Answer:

b) {1=A, 2=D, 3=C}

Explanation:

The `replace` method updates the value associated with key 2 to "D".

12. Identify the output of this code:

```java
List<Integer> list = new LinkedList<>();
list.add(1);
list.add(2);
list.add(3);
list.add(1, 4);
System.out.println(list);
```

a) [1, 4, 2, 3]

b) [4, 1, 2, 3]

c) [1, 2, 3, 4]

d) [1, 2, 4, 3]

Click to View Answer and Explanation

Answer:

a) [1, 4, 2, 3]

Explanation:

The `add` method with an index adds the element at the specified position, shifting others to the right.

13. What does this Java code snippet output?

```
Queue<String> queue = new PriorityQueue<>();
queue.offer("C");
queue.offer("A");
queue.offer("B");
System.out.println(queue.poll());
```

a) A

b) B

c) C

d) null

Click to View Answer and Explanation

Answer:

a) A

Explanation:

A `PriorityQueue` orders elements according to their natural ordering. "A" is polled first as it's the smallest.

14. What is the output of the following Java code?

```
List<Integer> list = new ArrayList<>();
for (int i = 1; i <= 5; i++) {
    list.add(i);
}
list.set(2, 10);
System.out.println(list);
```

a) [1, 2, 10, 4, 5]

b) [1, 10, 3, 4, 5]

c) [1, 2, 3, 10, 5]

d) [1, 2, 3, 4, 5]

Click to View Answer and Explanation

Answer:

a) [1, 2, 10, 4, 5]

Explanation:

The `set` method replaces the element at the specified index. Index 2 (third element) is changed from 3 to 10.

15. What will the following Java code snippet output?

```
Map<Integer, String> map = new LinkedHashMap<>();
map.put(3, "C");
map.put(1, "A");
map.put(2, "B");
for (Map.Entry<Integer, String> entry : map.entrySet()) {
    System.out.print(entry.getValue() + " ");
}
```

a) A B C

b) C A B

c) B C A

d) C B A

Click to View Answer and Explanation

Answer:

b) C A B

Explanation:

A `LinkedHashMap` maintains insertion order. The values are iterated in the order they were put into the map.

16. Identify the output of this code:

```
List<Integer> list = new CopyOnWriteArrayList<>(Arrays.asList(1, 2, 3));
for (Integer item : list) {
    if (item == 2) {
        list.remove(item);
    }
}
System.out.println(list);
```

a) [1, 2, 3]

b) [1, 3]

c) [2, 3]

d) [1, 2]

Click to View Answer and Explanation

Answer:

b) [1, 3]

Explanation:

`CopyOnWriteArrayList` allows safe removal during iteration. The element 2 is removed, leaving [1, 3].

17. What does this Java code snippet output?

```
Deque<Integer> deque = new ArrayDeque<>();
deque.addFirst(1);
deque.addFirst(2);
deque.addLast(3);
System.out.println(deque);
```

a) [1, 2, 3]

b) [2, 1, 3]

c) [3, 2, 1]

d) [3, 1, 2]

Click to View Answer and Explanation

Answer:

b) [2, 1, 3]

Explanation:

Elements are added to the front and back of the deque, resulting in [2, 1, 3].

18. What is the result of the following code snippet?

```
List<String> list = new Vector<>();
list.add("A");
list.add("B");
list.add("C");
System.out.println(list.contains("B"));
```

a) true

b) false

c) Compilation error

d) Runtime error

Click to View Answer and Explanation

Answer:

a) true

Explanation:

The `contains` method checks if the list contains the specified element. Since "B" is in the list, it returns `true` .

19. What will the following Java code snippet output?

```
SortedSet<String> sortedSet = new TreeSet<>();
sortedSet.add("C");
sortedSet.add("A");
sortedSet.add("B");
System.out.println(sortedSet.first());
```

a) A

b) B

c) C

d) null

Click to View Answer and Explanation

Answer:

a) A

Explanation:

A `TreeSet` sorts its elements. The `first` method returns the first (lowest) element, which is "A".

20. Identify the output of this code:

```
Map<Integer, String> map = new HashMap<>();
map.put(1, "One");
map.put(2, "Two");
map.put(3, "Three");
System.out.println(map.containsKey(2));
```

a) true

b) false

c) null

d) Compilation error

Click to View Answer and Explanation

Answer:

a) true

Explanation:

`containsKey` checks whether the map contains a mapping for the specified key. Since there is a key 2, it returns `true`.

21. What does this Java code snippet output?

```
Queue<Integer> queue = new PriorityQueue<>(Comparator.reverseOrder());
queue.offer(3);
queue.offer(1);
queue.offer(2);
System.out.println(queue.poll());
```

a) 1

b) 2

c) 3

d) null

Click to View Answer and Explanation

Answer:

c) 3

Explanation:

The `PriorityQueue` is initialized with a `Comparator` for reverse ordering, so it polls the largest element first, which is 3.

22. What is the result of executing this code?

```
List<Integer> list = new ArrayList<>(Arrays.asList(1, 2, 3));
Iterator<Integer> iterator = list.iterator();
while (iterator.hasNext()) {
    if (iterator.next() == 2) {
        iterator.remove();
    }
}
System.out.println(list);
```

a) [1, 2, 3]

b) [1, 3]

c) [2, 3]

d) [1, 2]

Click to View Answer and Explanation

Answer:

b) [1, 3]

Explanation:

The `Iterator` removes the element 2 from the list, leaving [1, 3].

23. What will the following Java code snippet output?

```
Set<String> set = new TreeSet<>(String.CASE_INSENSITIVE_ORDER);
set.add("apple");
set.add("Banana");
set.add("APPLE");
System.out.println(set);
```

a) [APPLE, Banana, apple]

b) [apple, Banana]

c) [Banana, APPLE, apple]

d) [apple, APPLE, Banana]

Click to View Answer and Explanation

Answer:

b) [apple, Banana]

Explanation:

The `TreeSet` is initialized with a case-insensitive order. It considers "apple" and "APPLE" as duplicates.

24. Identify the output of this code:

```
Map<Integer, String> map = new ConcurrentHashMap<>();
map.put(1, "A");
map.put(2, "B");
map.remove(1);
System.out.println(map);
```

a) {1=A, 2=B}

b) {2=B}

c) {1=A}

d) {}

Click to View Answer and Explanation

Answer:

b) {2=B}

Explanation:

The `remove` method removes the mapping for key 1, leaving {2=B}.

25. What does this Java code snippet output?

```
List<String> list = new LinkedList<>();
list.add("A");
list.add("B");
list.addFirst("C");
System.out.println(list);
```

a) [A, B, C]

b) [C, A, B]

c) [B, A, C]

d) [B, C, A]

Click to View Answer and Explanation

Answer:

b) [C, A, B]

Explanation:

The `addFirst` method adds the element at the beginning of the list, resulting in [C, A, B].

26. What is the result of the following code snippet?

```
Deque<String> deque = new LinkedList<>();
deque.offer("A");
deque.offerFirst("B");
deque.offerLast("C");
System.out.println(deque.poll());
```

a) A

b) B

c) C

d) null

Click to View Answer and Explanation

Answer:

b) B

Explanation:

`offerFirst` adds "B" at the beginning. `poll` retrieves and removes the first element, which is "B".

1. What is the result of executing this Java code snippet?

```
public class Test {
    public static void main(String[] args) {
        try {
            int[] numbers = {1, 2, 3};
            System.out.println(numbers[5]);
        } catch (ArrayIndexOutOfBoundsException e) {
            System.out.println("Array index out of bounds!");
        }
    }
}
```

a) The program prints "Array index out of bounds!"

b) The program prints a different error message

c) The program throws an `ArrayIndexOutOfBoundsException`

d) The program executes without any output

Click to View Answer and Explanation

Answer:

a) The program prints "Array index out of bounds!"

Explanation:

The code tries to access an index that is out of bounds for the array `numbers`. This triggers the `ArrayIndexOutOfBoundsException`, which is caught and handled in the `catch` block.

2. What does this Java code snippet output?

```
public class Test {
    public static void main(String[] args) {
        try {
            int x = 0;
            int y = 5 / x;
        } catch (Exception e) {
            System.out.println("Exception occurred");
        }
    }
}
```

a) 0

b) The program throws an `ArithmeticException`

c) Exception occurred

d) The program executes without any output

Click to View Answer and Explanation

Answer:

c) Exception occurred

Explanation:

Dividing by zero throws an `ArithmeticException`, which is caught by the `catch` block since it catches all `Exception` types.

3. Identify the output of the following code:

```
public class Test {
    public static void main(String[] args) {
        try {
            badMethod();
            System.out.println("A");
        } catch (Exception ex) {
            System.out.println("B");
        } finally {
            System.out.println("C");
        }
        System.out.println("D");
    }
    public static void badMethod() {
        throw new Error();
    }
}
```

a) A B C D

b) B C D

c) C

d) B C

Click to View Answer and Explanation

Answer:

c) C

Explanation:

The given code snippet throws an `Error` from the `badMethod()` method. It's important to note the difference between `Error` and `Exception` in Java:

`Error` represents serious problems that a reasonable application should not try to catch, while `Exception` is a condition that your application might want to catch.

Here's the flow of the code based on its behavior:

1. The main method calls `badMethod()`, which immediately throws an Error.

2. Because the thrown object is an `Error` and not an `Exception`, the catch block designed to catch an Exception will not catch this Error.

3. The `finally` block executes after the try block exits (whether the try block exits normally or abruptly due to an exception or error). So, "C" is printed.

4. After executing the `finally` block, since the Error is not caught by the catch block, it propagates up the call stack, and the rest of the code in the main method after the try-catch-finally block (i.e., System.out.println("D");) does not execute.

5. The program terminates abnormally due to the uncaught Error.

Therefore, the correct output is: **C**

4. What will be printed by this Java code?

```
public class Test {
    public static void main(String[] args) {
        try {
            System.out.println("Hello, world!");
        } finally {
            System.out.println("Finally executing...");
        }
    }
}
```

a) Hello, world!

b) Finally executing...

c) Hello, world! Finally executing...

d) The program throws an exception

Click to View Answer and Explanation

Answer:

c) Hello, world! Finally executing...

Explanation:

The `try` block is executed normally, and then the `finally` block is executed, resulting in both statements being printed.

5. What does this code snippet output?

```
public class Test {
    public static void main(String[] args) {
        try {
            String s = null;
            System.out.println(s.length());
        } catch (NullPointerException e) {
            System.out.println("Caught NullPointerException");
        } catch (Exception e) {
            System.out.println("Caught Exception");
        }
    }
}
```

a) Caught NullPointerException

b) Caught Exception

c) An uncaught exception is thrown

d) The program executes without any output

Click to View Answer and Explanation

Answer:

a) Caught NullPointerException

Explanation:

Accessing the length of a `null` string throws a `NullPointerException`, which is caught by the first `catch` block.

6. What is the result of executing this code?

```
public class Test {
    public static void main(String[] args) {
        try {
            int[] arr = new int[-5];
        } catch (NegativeArraySizeException e) {
            System.out.println("Negative array size");
        }
    }
}
```

a) Negative array size

b) An array with size -5 is created

c) The program throws a different type of exception

d) The program executes without any output

Click to View Answer and Explanation

Answer:

a) Negative array size

Explanation:

Trying to create an array with a negative size throws a `NegativeArraySizeException`, which is caught and handled.

7. What will the following Java code snippet output?

```
public class Test {
    public static void main(String[] args) {
        try {
            throw new RuntimeException();
        } catch (RuntimeException e) {
            System.out.println("RuntimeException caught");
        } catch (Exception e) {
            System.out.println("Exception caught");
        }
    }
}
```

a) RuntimeException caught

b) Exception caught

c) The program throws an uncaught exception

d) The program executes without any output

Click to View Answer and Explanation

Answer:

a) RuntimeException caught

Explanation:

The `RuntimeException` is caught by the first `catch` block that matches its type.

8. What does the following code snippet print?

```
public class Test {
    public static void main(String[] args) {
        String str = null;
        try {
            System.out.println(str.length());
        } catch (NullPointerException e) {
            System.out.println("Null Pointer Exception");
        } finally {
            System.out.println("Finally block executed");
        }
    }
}
```

a) Null Pointer Exception

b) Finally block executed

c) Null Pointer Exception

d) The program throws an uncaught exception

Click to View Answer and Explanation

Answer:

c) Null Pointer Exception Finally block executed

Explanation:

A `NullPointerException` is thrown and caught, then the `finally` block executes, resulting in both messages being printed.

9. Determine the output of this Java code:

```
class MyException extends Exception {}

public class Test {
    public static void main(String[] args) {
        try {
            throw new MyException();
        } catch (MyException e) {
            System.out.println("Custom exception caught");
        }
    }
}
```

a) Custom exception caught

b) The program throws an uncaught exception

c) Compilation error

d) Runtime error

Click to View Answer and Explanation

Answer:

a) Custom exception caught

Explanation:

`MyException` is a custom exception class. It's thrown and immediately caught in the `catch` block.

10. What is the result of the following code snippet?

```
public class Test {
    public static void main(String[] args) {
        try {
            riskyMethod();
        } catch (Exception e) {
            System.out.println("Exception caught");
        }
    }
    static void riskyMethod() throws Exception {
        throw new Exception("Problem");
    }
}
```

a) Exception caught

b) A message "Problem" is printed

c) The program throws an uncaught exception

d) Compilation error

Click to View Answer and Explanation

Answer:

a) Exception caught

Explanation:

`riskyMethod` throws an `Exception` with the message "Problem", which is caught by the `catch` block in the `main` method.

1. What is the output of the following code snippet?

```
int x = 5;
int y = 10;
System.out.println(x + y + "Hello");
```

a) 15Hello

b) Hello15

c) 510Hello

d) 5Hello10

Click to View Answer and Explanation

Answer:

a) 15Hello

Explanation:

In Java, when the `+` operator is used with a combination of numbers and strings, it performs addition for numbers and concatenation for strings. The expression x + y evaluates to 15, and then "Hello" is concatenated to it. Therefore, the final output is **15Hello**.

2. What is the output of the following code snippet?

```
public class Main{
     public static void main(String []args){
        String str1 = "Hello";
        String str2 = new String("Hello");
        System.out.println(str1 == str2);
     }
}
```

a) true

b) false

c) It will cause a compilation error.

d) It will throw a runtime exception.

Click to View Answer and Explanation

Answer:

b) false

Explanation:

The `==` operator compares the object references. In this case, `str1` and `str2` refer to different objects, even though they contain the same string value. Therefore, the output is `false`.

3. What is the output of the following code snippet?

```
int[] arr = {1, 2, 3, 4, 5};
System.out.println(arr.length);
```

a) 1

b) 2

c) 3

d) 5

Click to View Answer and Explanation

Answer:

d) 5

Explanation:

The `length` property of an array in Java returns the number of elements in the array. In this case, the array `arr` has 5 elements, so the output is 5.

4. What is the output of the following Java program?

```java
public class Main{
     public static void main(String []args){
        int x = 5;
        int y = 10;
        if (x < y)
            System.out.println("x is less than y");
        else if (x > y)
            System.out.println("x is greater than y");
        else
            System.out.println("x is equal to y");
     }
}
```

a) x is less than y

b) x is greater than y

c) x is equal to y

d) The code will not compile due to a syntax error.

Click to View Answer and Explanation

Answer:

a) x is less than y

Explanation:

The if condition `x < y` is true, so the code inside the first if block is executed, resulting in the output x is less than y

5. What is the output of the following Java program?

```java
public class Main{
     public static void main(String []args){
        int x = 10;
        int y = 5;
        int z = (x > y) ? x : y;
        System.out.println(z);
      }
}
```

a) 10

b) 5

c) 15

d) The code will not compile due to a syntax error.

Click to View Answer and Explanation

Answer:

a) 10

Explanation:

The ternary operator `(x > y) ? x : y` evaluates to x because x is greater than y. Therefore, the output is 10.

6. What is the output of the following Java program?

```
public class Main{
     public static void main(String []args){
        int x = 10;
        while (x > 0) {
            System.out.print(x + " ");
            x--;
        }
     }
}
```

a) 10 9 8 7 6 5 4 3 2 1

b) 10 9 8 7 6 5 4 3 2 1 0

c) 1 2 3 4 5 6 7 8 9 10

d) The code will not compile due to a syntax error.

Click to View Answer and Explanation

Answer:

a) 10 9 8 7 6 5 4 3 2 1

Explanation:

The while loop iterates as long as the condition x > 0 is true. In each iteration, the value of x is printed, and then it is decremented. Therefore, the output is "10 9 8 7 6 5 4 3 2 1".

7. What is the output of the following Java program?

```
public class Main{
     public static void main(String []args){
        for (int i = 0; i < 5; i++) {
            System.out.print(i + " ");
            if (i == 2)
                break;
        }
     }
}
```

a) 0 1 2

b) 0 1 2 3 4

c) 0 1

d) 0 1 2 3 4 5

Click to View Answer and Explanation

Answer:

a) 0 1 2

Explanation:

The for loop iterates from 0 to 4. When `i` becomes 2, the if condition `i == 2` is true, and the break statement is executed, terminating the loop. Therefore, the output is "0 1 2".

8. What is the output of the following Java program?

```
public class Main{
     public static void main(String []args){
        for (int i = 0; i < 5; i++) {
            if (i == 2)
                continue;
            System.out.print(i + " ");
        }
     }
}
```

a) 0 1 2 3 4

b) 0 1 3 4

c) 2

d) 0 1 3 4 5

Click to View Answer and Explanation

Answer:

b) 0 1 3 4

Explanation:

The for loop iterates from 0 to 4. When `i` becomes 2, the if condition `i == 2` is true, and the continue statement is executed, skipping the rest of the loop body for that iteration. Therefore, the output is "0 1 3 4".

9. What is the output of the following Java program?

```
public class Main{
     public static void main(String []args){
        int i = 0;
        do {
            System.out.print(i + " ");
            i++;
        } while (i < 5);
```

```
    }
}
```

a) 0 1 2 3 4

b) 0 1 2 3 4 5

c) 1 2 3 4 5

d) The code will not compile due to a syntax error.

Click to View Answer and Explanation

Answer:

a) 0 1 2 3 4

Explanation:

The do-while loop executes the loop body at least once and continues as long as the condition i < 5 is true. Therefore, the output is "0 1 2 3 4".

10. What is the output of the following program?

```
public class Main {
    public static void main(String[] args) {
        int x = 10;
        int y = x++;
        System.out.println(y);
    }
}
```

a) 10

b) 11

c) 9

d) Compile-time error

Click to View Answer and Explanation

Answer:

a) 10

Explanation:

The postfix increment operator (x++) first assigns the value of x to y, and then increments the value of x. Therefore, the value of y is 10.

11. What will be the output of the following Java program?

```
class Base {
    public Base() {
        System.out.println("Base");
    }
}

class Derived extends Base {
    public Derived() {
        System.out.println("Derived");
    }
}

class DeriDerived extends Derived {
    public DeriDerived() {
        System.out.println("DeriDerived");
    }
}

public class Test {
    public static void main(String[] args) {
        Derived b = new DeriDerived();
    }
}
```

a)

```
Base
Derived
DeriDerived
```

b)

```
Derived
DeriDerived
```

c)

```
DeriDerived
Derived
Base
```

d)

```
DeriDerived
Derived
```

Click to View Answer and Explanation

Answer:

a)

```
Base
Derived
DeriDerived
```

Explanation:

Whenever a class gets instantiated, the constructor of its base classes (the constructor of the root of the hierarchy gets executed first) gets invoked before the constructor of the instantiated class.

12. What will be the output of the following Java program?

```
public class Test {
    public void print(Integer i) {
        System.out.println("Integer");
    }

    public void print(int i) {
        System.out.println("int");
    }

    public void print(long i) {
        System.out.println("long");
    }

    public static void main(String args[]) {
        Test test = new Test();
        test.print(10);
    }
}
```

a) The program results in a compiler error (“ambiguous overload”).

b) long

c) Integer

d) int

Click to View Answer and Explanation

Answer:

d) int

Explanation:

For an integer literal, the JVM matches in the following order: **int, long, Integer, int....** In other words, it first looks for an `int` type parameter; then it looks for `long` type; and so on. Here, since the `int` type parameter is specified with an overloaded method, it matches with `int` .

13. What will be the output of the following Java program?

```
public class StrEqual {
    public static void main(String[] args) {
        String s1 = "hello";
        String s2 = new String("hello");
        String s3 = "hello";
        if (s1 == s2) {
            System.out.println("s1 and s2 equal");
        } else {
            System.out.println("s1 and s2 not equal");
        }
        if (s1 == s3) {
            System.out.println("s1 and s3 equal");
        } else {
            System.out.println("s1 and s3 not equal");
        }
    }
}
```

Which one of the following options provides the output of this program when executed?
a)

```
s1 and s2 equal
s1 and s3 equal
```

b)

```
s1 and s2 equal
s1 and s3 not equal
```

c)

```
s1 and s2 not equal
s1 and s3 equal
```

d)

```
s1 and s2 not equal
s1 and s3 not equal
```

Click to View Answer and Explanation

Answer:

c)

```
s1 and s2 not equal
s1 and s3 equal
```

Explanation:

JVM sets a constant pool in which it stores all the string constants used in the type. If two references are declared with a constant, then both refer to the same constant object. The `==` operator checks the similarity of the objects themselves (and not the values in it). Here, the first comparison is between two distinct objects, so we get `s1` and `s2` not equal. On the other hand, since references to `s1` and `s3` refer to the same object, we get `s1` and `s3` equal.

14. What is the output of the following Java program?

```
public class Test {

    public static void main(String[] args) {
        String s1 = "hello";
        String s2 = new String("hello");

        s2 = s2.intern();
        System.out.println(s1 == s2);
    }
}
```

a) false

b) true

Click to View Answer and Explanation

Answer:

b) true

Explanation:

We know that the `intern()` method will return the String object reference from the string pool since we assign it back to `s2` and now both `s1` and `s2` are having the same reference. It means that `s1` and `s2` references point to the same object.

15. What is the output of the following code snippet?

```
public class Main {
    public static void main(String[] args) {
        String str = "Java";
        str.concat(" Programming");
        System.out.println(str);
    }
}
```

a) Java

b) Java Programming

c) Programming

d) Compile-time error

Click to View Answer and Explanation

Answer:

a) Java

Explanation:

The `concat()` method returns a new string resulting from concatenating the specified string to the original string. However, in this code, the result of concat() is not assigned back to the str variable. Therefore, the original string "Java" remains unchanged, and the output is "Java".

16. What is the output of the following code snippet?

```java
import java.util.regex.*;

public class RegexQuiz {
    public static void main(String[] args) {
        String regex = "\\d+";
        String input = "1234";

        Pattern pattern = Pattern.compile(regex);
        Matcher matcher = pattern.matcher(input);

        while (matcher.find()) {
            System.out.print(matcher.group() + " ");
        }
    }
}
```

a) 1

b) 1234

c) 123456789

d) Compile-time error

Click to View Answer and Explanation

Answer:

b) 1234

Explanation:

The regex pattern "\\d+" is used, which matches one or more digits. Since the input string "1234" consists of only digits, the regex pattern matches the entire string. Therefore, the output will be "1234".

17. What is the output of the following code snippet?

```java
import java.util.regex.*;

public class RegexQuiz {
    public static void main(String[] args) {
        String regex = "[a-c]";
        String input = "abcABC";

        Pattern pattern = Pattern.compile(regex);
        Matcher matcher = pattern.matcher(input);

        while (matcher.find()) {
            System.out.print(matcher.group() + " ");
        }
    }
}
```

a) a b c

b) A B C

c) a b c A B C

d) Compile-time error

Click to View Answer and Explanation

Answer:

a) a b c

Explanation:

The regex "[a-c]" matches any character between 'a' and 'c' (inclusive) in a case-sensitive manner. In the input string "abcABC", it matches "a", "b", and "c" and prints them.

18. What is the output of the following Java program?

```
class One{
        public static void print(){
                System.out.println("1");
        }
}

class Two extends One{
        public static void print(){
                System.out.println("2");
        }
}

public class Test{
        public static void main(String args[]){
                One one = new Two();
                one.print();
        }
}
```

a) 2

b) 1

c) Compile-time error

d) Run-time error

Click to View Answer and Explanation

Answer:

b) 1

Explanation:

Static methods are defined at the class level. In the case of the static methods, regardless of which object the reference is pointing to, it always calls the static method defined by the reference class. Here, the reference is of type `One` , so the static method of class `One` will be called.

19. What is the output of the following Java program?

```
class One{
        public void print(){
                System.out.println("1");
        }
}

class Two extends One{
        public void print(){
                System.out.println("2");
        }
}

public class Test{
        public static void main(String args[]){
                One one = new Two();
                one.print();
        }
}
```

a) 2

b) 1

c) Compile-time error

d) Run-time error

Click to View Answer and Explanation

Answer:

a) 2

Explanation:

Even though the reference type of object one is `One`, the actual object it points to is of type `Two` due to polymorphism. When we call the print method on `One`, the JVM dynamically binds the call to the print method in class `Two` at runtime. As a result, "2" is printed on the console.

20. What is the output of the following Java program?

```
class One{

        public One(int x){
                System.out.print("int constructor");
        }

        public One(long l){
                System.out.print("long constructor");
        }
}

public class Test{

        public static void main(String[] args){
                long l = 20L;
                One one = new One(l);
        }
}
```

a) int constructor

b) long constructor

c) Compile-time error

d) Run-time error

Click to View Answer and Explanation

Answer:

b) long constructor

Explanation:

When we create object `one` with the `long` variable l as an argument, the constructor with the long parameter in class `One` is called. As a result, the "long constructor" is printed to the console.
If you pass an `int` parameter in class `One` then it will print "int constructor" to the console.

21. What is the output of the following Java program?

```
class Parent{
        public void className(){
                System.out.println("Parent");
        }
}
class Child extends Parent{
        void className(){
                System.out.println("Child");
        }
}

public class Test{

        public static void main(String[] args){
                Parent parent = new Child();
                parent.className();
        }
}
```

a) Parent

b) Child

c) Compile-time error

d) Run-time error

Click to View Answer and Explanation

Answer:

c) Compile-time error

Explanation:

When overriding a parent class method in a child class, we cannot reduce the visibility of the method. For example, if the method is defined as public in the parent class, a child class cannot override it with protected. The code will give the compilation error **“Cannot reduce the visibility of the inherited method from Parent”.**

22. What is the output of the following Java program?

```
class Demo{
        public Demo(int i){
                System.out.println("int");
        }

        public void Demo(short s){
                System.out.println("short");
        }
}

public class Test{

        public static void main(String[] args){
                short s = 10;
                Demo demo = new Demo(s);
        }
}
```

a) int

b) short

c) Compile-time error

d) Run-time error

Click to View Answer and Explanation

Answer:

a) int

Explanation:

The class `Demo` has one constructor i.e. with `int` argument. The `short` value is automatically promoted to an `int` value during object creation so the constructor with the `int` argument will be called and it will print "int".

23. What is the output of the following Java program?

```
class Demo{
        void Demo(){
                System.out.println("Demo");
        }

}

public class Test{

        public static void main(String[] args){
                Demo demo = new Demo();
        }
}
```

a) Demo

b) No Output

c) Compile-time error

d) Run-time error

Click to View Answer and Explanation

Answer:

b) No Output

Explanation:

Java constructors do not have a return type. If you mention the return type, it automatically becomes a method instead of a constructor. Hence, the `void Demo()` becomes a method of the class `Demo` and thus it will not be called when an object of the class `Demo` is created.

24. What is the output of the following Java program?

```
class One{
        public One(){
                System.out.print("One,");
        }
}
class Two extends One{
        public Two(){
                System.out.print("Two,");
        }
}
class Three extends Two{
        public Three(){
                System.out.print("Three");
        }
}

public class Test{

        public static void main(String[] args){
                Three three = new Three();
        }
}
```

a) Three

b) One

c) One,Two,Three

d) Run-time error

Click to View Answer and Explanation

Answer:

c) One,Two,Three

Explanation:

When we create an object of class `Three` , the constructors are executed in the following order:
One(): Prints "One,".
Two(): Prints "Two,".
Three(): Prints "Three".
So, the overall output is "One,Two,Three". The constructors are executed in the order of inheritance hierarchy from the topmost superclass (One) to the subclass (Three).

25. What is the output of the following Java program?

```
class Hello{
        public Hello(){
                System.out.println("Hello");
        }
}

public class Main{

        Hello hello = new Hello();

        public static void main(String[] args) {
                System.out.println("Hello World!");
        }
}
```

a) Hello

b) Hello World!

c) No Output

d) Run-time error

Click to View Answer and Explanation

Answer:

b) Hello World!

Explanation:

The execution of the Java program starts from the `main()` method so it will print "Hello World!"

26. What is the output of the following Java program?

```
public class Main{

        static String name = "Ramesh";

        public Main(){
                name = "Prabhas";
        }

        public static void main(String[] args){
                System.out.println("The name is " + name);
        }
}
```

a) Prabhas

b) The name is Ramesh

c) No Output

d) Run-time error

Click to View Answer and Explanation

Answer:

b) The name is Ramesh

Explanation:

The `name` String variable is declared as static and initialized with the string “Ramesh”. The value of the `name` variable is changed in the constructor. However, the class constructor is called only when an object is created. The code does not create any objects of the class Test, and hence the constructor is never called. So the value of the `name` variable remains unchanged.

27. What will be the output of the following program?

```
class First
{
    static void staticMethod()
    {
        System.out.println("Static Method");
    }
}

public class MainClass
{
    public static void main(String[] args)
    {
        First first = null;

        first.staticMethod();
    }
}
```

a) Static Method

b) Throws a NullPointerException

c) Compile-time error

d) Runtime error

Click to View Answer and Explanation

Answer:

a) Static Method

Explanation:

The provided Java code will compile and execute successfully without any exceptions. When calling a static method, it doesn't require an instance of the class. Therefore, you can call the static method `staticMethod()` from class `First` using the null reference `first` .

https://www.youtube.com/@codewitharrays

https://www.instagram.com/codewitharrays/

https://t.me/codewitharrays Group Link: https://t.me/ccee2025notes

+91 8007592194 +91 9284926333

codewitharrays@gmail.com

https://codewitharrays.in/project