Maveric Questions

Find the oputput:

1. **package** org.sample;

**public** **class** OutputSample {

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

**int** arr[]= {1,2,3,4,5,6,7,8};

**for** (**int** i = 0; i < arr.length; i++) {

**if**(arr[i]>=5) {

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

System.*exit*(0);

}

**else**

System.***out***.println("arr["+i+"]="+arr[i]);

}

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

}

}

Output:

arr[0]=1

arr[1]=2

arr[2]=3

arr[3]=4

exit

2. **public** **class** OutputSample {

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

**try** {

**int** x=0;

**int** y=5;

**int** z=y/x;

} **catch** (Exception e) {

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

}

**catch**(ArithmeticException ae) {

System.***out***.println("arithmetic exception");

}

}

}

Output:

Exception in thread "main" java.lang.Error: Unresolved compilation problem:

Unreachable catch block for ArithmeticException. It is already handled by the catch block for Exception

at org.sample.OutputSample.main(OutputSample.java:13)

3. **public** **class** OutputSample {

**static** **int** *count*=0;

**private** **void** increment() {

*count*++;

}

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

OutputSample o=**new** OutputSample();

OutputSample o1=**new** OutputSample();

o.increment();

o1.increment();

System.***out***.println("o : count is= "+o.*count*);

System.***out***.println("o1 : count is= "+o1.*count*);

}

}

Output:

o : count is= 2

o1 : count is= 2

4. **package** org.sample;

**public** **abstract** **class** Base {

**public** Base() {

System.***out***.println("base constructor called");

}

**abstract** **void** fun();

}

**package** org.sample;

**public** **class** Derived **extends** Base{

**public** Derived() {

System.***out***.println("derived constructor");

}

@Override

**void** fun() {

System.***out***.println("derived fun() called");

}

}

**package** org.sample;

**public** **class** Main {

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

Derived d=**new** Derived();

}

}

Output:

base constructor called

derived constructor

5. **package** org.sample;

**public** **class** Base {

**public** **void** transact() {

System.***out***.println("Parent :: inside transact");

}

}

**package** org.sample;

**public** **class** Derived **extends** Base{

@Override

**public** **void** transact() {

System.***out***.println("child :: inside transact");

}

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

Base o=**new** Derived();

o.transact();

}

}

Output:

child :: inside transact

Programs:

1.Remove all adjacent matching characters from String

**package** org.sample;

**public** **class** OutputSample {

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

String s="AABBCBBC";

String re="";

**for** (**int** i = 0; i <s.length()-1; i++) {

**char** c = s.charAt(i);

**char** c1 = s.charAt(i+1);

**if**(c!=c1) {

re=re+c;

}

}

System.***out***.println(re+s.charAt(s.length()-1));

}

}

Output:

ABCBC

2.Create a excel for following data and store it in desktop

|  |  |
| --- | --- |
| Country | Captial |
| India | NewDelhi |
| England | London |

**public** **class** AppTest {

**public** **static** **void** main(String[] args) **throws** IOException {

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

File f= **new** File("C:\\Users\\User\\Desktop\\Write.xlsx");

Workbook w = **new** XSSFWorkbook();

Sheet s = w.createSheet("test");

**for** (**int** i = 0; i <=2; i++) {

Row r = s.createRow(i);

**for** (**int** j = 0; j <=1; j++) {

Cell c = r.createCell(j);

String s2 = sc.next();

c.setCellValue(s2);

}

}

FileOutputStream o=**new** FileOutputStream(f);

w.write(o);

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

}

}

3.Occurance of character in given String

**public** **class** Main {

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

Map<Character,Integer> mp=**new** LinkedHashMap<Character,Integer>();

String s="MAVERIC\_CHENNAI";

**for** (**int** i = 0; i < s.length(); i++) {

**char** c = s.charAt(i);

**if**(mp.containsKey(c)) {

Integer k = mp.get(c);

mp.put(c, k+1);

}

**else** {

mp.put(c, 1);

}

}

System.***out***.println(mp);

}

}

4.Write a selenium script to read content from Web table and find the total cost of all unique product

|  |  |
| --- | --- |
| Knife | 20 |
| Cup | 10 |
| Spoon | 5 |
| Knife | 25 |
| Pan | 40 |
| Cup | 5 |

**public** **class** OutputSample {

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

**int** total=0;

Map<String,Integer> mp=**new** LinkedHashMap<>();

System.*setProperty*("webdriver.chrome.driver", "C:\\Users\\User\\eclipse\\Demo123\\Maveric\\Driver\\chromedriver.exe");

WebDriver driver = **new** ChromeDriver();

driver.get("https://www.guru99.com/handling-dynamic-selenium-webdriver.html");

List<WebElement> tr = driver.findElements(By.*tagName*("tr"));

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

String tx = tr.get(i).getText();

String[] sp = tx.split(" ");

String s = sp[0];

String s1 = sp[1];

**int** j = Integer.*parseInt*(s1);

**if**(mp.containsKey(s)) {

Integer k = mp.get(s);

total=total+k;

mp.put(s, total);

}

**else** {

mp.put(s, total);

}

}

System.***out***.println(mp);

}

}