|  |  |
| --- | --- |
| **Question 1 of 4** | 2.5 Points |

**For:**

**class Something{**

**int[] someValues = {10,20,30};**

**}**

**public class Main {**

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

**Something [] some = new Something[3];**

**some[0] = new Something();**

**Something aThing = new Something();**

**some[1] = aThing;**

**aThing = null;**

**some[1] = null;**

**//other things**

**// The following lines will show the 5 objects created with “new”**

**// System.out.println("some[0]: "+some[0]); //** Something@15db9742

**// System.out.println("some[1]: "+some[1]); //** null

**// System.out.println("some[2]: "+some[2]); //** null

**// System.out.println("aThing: "+aThing); //** null

**// The fact that some of them will have reference to other and will get to GC is not relevant for the answer. The question was “How many objects ARE created”**

**}**

**}**

**how many objects are created ?**

|  |  |  |  |
| --- | --- | --- | --- |
| |  | | --- | |  | | A. 4 |  |
| |  | | --- | |  | | B. 5 |  |
| |  | | --- | |  | | C. 3 |  |
| |  | | --- | |  | | D. 1 |  |
| |  | | --- | |  | | E. 2 |  |

|  |  |
| --- | --- |
| **Question 2 of 4** | 2.5 Points |

**Given:**

**class Container{**

**private String name;**

**private static Container instance = null;**

**private Container(){**

**this.name = "Nothing";**

**}**

**public static Container getInstance(){**

**if (instance == null){**

**instance = new Container();**

**}**

**return instance;**

**}**

**public void setName(String x) { this.name = x; }**

**public String getName() { return this.name; }**

**}**

**public class Main{**

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

**Container s1 = Container.getInstance(); // creates s1 in heap**

**Container s2 = Container.getInstance(); // it won’t create another object because getInstance won’t return null and will return the s1 reference that was created before; s1 and s2 will point to the same object in heap**

**s1.setName("Container 1"); // change name of s1 from “Nothing” to “Container 1”**

**s2.setName("Container 2"); // this changed the name of s1 object to “Container 2” because s2 has the same reference as s1 and will change the same object in heap**

**System.out.println("s1=" + s1.getName() + " s2=" + s2.getName());**

**}**

**}**

**what is printed ?**

|  |  |  |  |
| --- | --- | --- | --- |
| |  | | --- | |  | | A. s1=Container 1 s2=Container 1 |  |
| |  | | --- | |  | | B. compiler errors because the constructor is private |  |
| |  | | --- | |  | | C. s1=Container 2 s2=Container 2 |  |
| |  | | --- | |  | | D. s1=Container 1 s2=Container 2 |  |
| |  | | --- | |  | | E. s1=Container 2 s2=Container 1 |  |
| |  | | --- | |  | | F. s1=Nothing s2=Nothing |  |

|  |  |
| --- | --- |
| **Question 3 of 4** | 2.5 Points |

**class Base{**

**public int vb;**

**public void Test(){System.out.print(" Test 1");}**

**}**

**class Subclass extends Base{**

**public void Test(){System.out.print(" Test 2");}**

**}**

**public class Main {**

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

**Base b = new Base(); // create Base object**

**b.Test(); // print “Test 1”**

**Subclass d = new Subclass(); // create Subclass object**

**d.Test(); // print “Test 2”**

**b = d; // b will change reference to d**

**b.Test(); // print “Test 2”**

**}**

**}**

|  |
| --- |
|  |

|  |  |  |  |
| --- | --- | --- | --- |
| |  | | --- | |  | | A. prints Test 1 Test 2 Test 2 |  |
| |  | | --- | |  | | B. prints Test 1 Test 1 Test 1 |  |
| |  | | --- | |  | | C. prints Test 1 Test 2 Test 1 |  |
| |  | | --- | |  | | D. the example generates a ClassCastException because you can NOT do Downcasting |  |
| |  | | --- | |  | | E. prints Test 2 Test 2 Test 1 |  |

|  |  |
| --- | --- |
| **Question 4 of 4** | 2.5 Points |

**class Test {**

**public static String s = "\*";**

**void s1()**

**{**

**try{**

**s2();**

**}**

**catch (Exception e){**

**s += "c";**

**}**

**}**

**void s2() throws Exception {**

**s3();**

**s+= "2";**

**s3();**

**s+="2b";**

**}**

**void s3() throws Exception{**

**throw new Exception();**

**}**

**}**

**public class Main {**

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

**new Test().s1(); // create object with s=”\*” and call s1**

**// s1 will try to call s2**

**// s2 will throw exception to s1**

**// s1 will catch the exception**

**// and will add “c” to the end of s**

**// s will become “\*c”**

**System.out.println(Test.s);**

**}**

**}**

**what is the result ?**

|  |  |  |  |
| --- | --- | --- | --- |
| |  | | --- | |  | | A. \*c2 |  |
| |  | | --- | |  | | B. \*c22b |  |
| |  | | --- | |  | | C. \*2c2b |  |
| |  | | --- | |  | | D. \* |  |
| |  | | --- | |  | | E. \*c |  |
| |  | | --- | |  | | F. \*2c |  |