**Question 1**

Suppose that an instance of class C has legal implementations of the hashCode and equals methods. Within any one execution of the Java application, the hash code contract requires that each invocation of the hashCode method on the same instance of class C must consistently return the same result as long as the fields used for the equals comparison remain unchanged.

|  |  |
| --- | --- |
| a. | false |
| b. | true |

**Question 2**

If two instances of a class type are equal according to the equals method, then the same integer value must be returned by the hashCode method of the two objects.

|  |  |
| --- | --- |
| a. | false |
| b. | true |

**Question 3**

If two instances of a class type are not equal according to the equals method, then the same integer value must not be returned by the hashCode method of the two objects.

|  |  |
| --- | --- |
| a. | false |
| b. | true |

**Question 4**

class A {

static void m1 (B a, B b, B c, B d, B e, B f, B g, B h) {

if (a.equals(b)) {System.out.print("A");}

if (!c.equals(d)) {System.out.print("B");}

if (e.hashCode() == f.hashCode()) {System.out.print("C");}

if (g.hashCode() != h.hashCode()) {System.out.print("D");}

}}

Suppose that method m1 is invoked with eight instances of the same class and the output is ABCD. If the B.equals andB.hashCode methods are implemented according to the hash code contract, then which of the following statements must always be true?

|  |  |
| --- | --- |
| a. | (a.hashCode() == b.hashCode()) |
| b. | (c.hashCode() != d.hashCode()) |
| c. | (e.equals(f)) |
| d. | (!g.equals(h)) |

**Question 5**

class B {

private int i1;

public int hashCode() {return 1;}

}

class C {

private int i1;

public int hashCode() {return -1;}

}

class D {

private int i1;

public int hashCode() {return i1;}

}

Suppose that the equals method of classes B, C and D all make use of the value of the int variable, i1. Which class has ahashCode method that is not consistent with the hash code contract?

|  |  |
| --- | --- |
| a. | B |
| b. | C |
| c. | D |
| d. | None of the above |

**Question 6**

Which of the following classes override both the equals and hashCode methods?

|  |  |
| --- | --- |
| a. | java.lang.Byte |
| b. | java.lang.Integer |
| c. | java.util.Vector |
| d. | java.lang.String |
| e. | java.lang.StringBuffer |

**Question 7**

class A {

int i1, i2;

public void setI1(int i) {i1 = i;}

public int getI1() {return i1;}

public void setI2(int i) {i2 = i;}

public int getI2() {return i2;}

public A(int ii1, int ii2) {i1 = ii1; i2 = ii2;}

public boolean equals(Object obj) {

if (obj instanceof A) {

return (i1 == ((A)obj).getI1());

}

return false;

}

public int hashCode() {

// Insert statement here.

}}

Which of the following statements could be inserted at the specified location without violating the hash code contract?

|  |  |
| --- | --- |
| a. | return 31; |
| b. | return getI1(); |
| c. | return getI2(); |
| d. | return 31 \* getI1() + getI2(); |

**Question 8**

class A {

int i1, i2;

public void setI1(int i) {i1 = i;}

public int getI1() {return i1;}

public void setI2(int i) {i2 = i;}

public int getI2() {return i2;}

public A(int ii1, int ii2) {i1 = ii1; i2 = ii2;}

public boolean equals(Object obj) {

if (obj instanceof A) {

return (i1 == ((A)obj).getI1()) & (i2 == ((A)obj).getI2());

}

return false;

}

public int hashCode() {

// Insert statement here.

}}

If inserted at the specified location, which of the following statements would produce the most efficient hashCode method?

|  |  |
| --- | --- |
| a. | return 31; |
| b. | return getI1(); |
| c. | return getI2(); |
| d. | return getI1() + getI2(); |
| e. | return 31 \* getI1() + getI2(); |
| f. | None of the above |