**Question 1**

Which of the following are true statements?

|  |  |
| --- | --- |
| a. | Encapsulation is a form of data hiding. |
| b. | A tightly encapsulated class is always immutable. |
| c. | Encapsulation is always used to make programs run faster. |
| d. | Encapsulation helps to protect data from corruption. |
| e. | Encapsulation allows for changes to the internal design of a class while the public interface remains unchanged. |

**Question 2**

Which of the following are true statements?

|  |  |
| --- | --- |
| a. | A top-level class can not be called "tightly encapsulated" unless it is declared private. |
| b. | Encapsulation enhances the maintainability of the code. |
| c. | A tightly encapsulated class allows fast public access to member fields. |
| d. | A tightly encapsulated class allows access to data only through accessor and mutator methods. |
| e. | Encapsulation usually reduces the size of the code. |
| f. | A tightly encapsulated class might have mutator methods that validate data before it is loaded into the internal data model. |

**Question 3**

A class can not be called "tightly encapsulated" unless which of the following is true?

|  |  |
| --- | --- |
| a. | The class is declared final. |
| b. | All local variables are declared private. |
| c. | All method parameters are declared final. |
| d. | No method returns a reference to any object that is referenced by an internal data member. |
| e. | None of the above |

**Question 4**

A class can not be called "tightly encapsulated" unless which of the following is true?

|  |  |
| --- | --- |
| a. | All of the methods are declared private. |
| b. | All of the methods are synchronized. |
| c. | All local variables are declared final. |
| d. | The class is a direct subclass of Object. |
| e. | Accessor methods are used to prevent fields from being set with invalid data. |
| f. | None of the above |

**Question 5**

A class can not be called "tightly encapsulated" unless which of the following are true?

|  |  |
| --- | --- |
| a. | The data members can not be directly manipulated by external code. |
| b. | The class is declared final. |
| c. | It has no public mutator methods. |
| d. | The superclass is tightly encapsulated. |

**Question 6**

A class can not be called "tightly encapsulated" unless which of the following is true?

|  |  |
| --- | --- |
| a. | The class is a nested class. |
| b. | The constructors are declared private. |
| c. | The mutator methods are declared private. |
| d. | The class implements the Encapsulated interface. |
| e. | None of the above |

**Question 7**

A class can not be called "tightly encapsulated" unless which of the following is true?

|  |  |
| --- | --- |
| a. | All member fields are declared final. |
| b. | The class is not anonymous. |
| c. | The internal data model can be read and modified only through accessor and mutator methods. |
| d. | The class is an inner class. |
| e. | None of the above |

**Question 8**

class GFC500 {private String name;}

class GFC501 {

private String name;

private void setName(String name) {this.name = name;}

private String getName() {return name;}

}

class GFC502 {

private String name;

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

public String getName() {return name;}

}

Which class is not tightly encapsulated?

|  |  |
| --- | --- |
| a. | GFC501 |
| b. | GFC502 |
| c. | GFC503 |
| d. | None of the above |

**Question 9**

class GFC505 extends GFC504 {

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

public String getName() {return name;}

}

class GFC504 extends GFC503 {

private void setName(String name) {this.name = name;}

private String getName() {return name;}

}

class GFC503 {String name;}

Which class is tightly encapsulated?

|  |  |
| --- | --- |
| a. | GFC503 |
| b. | GFC504 |
| c. | GFC505 |
| d. | None of the above |

**Question 10**

class GFC506 {private String name;}

class GFC507 extends GFC506 {

String name;

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

public String getName() {return name;}

}

class GFC508 extends GFC506 {

private String name;

public GFC508(String name) {setName(name);}

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

public String getName() {return name;}

}

Which class is not tightly encapsulated?

|  |  |
| --- | --- |
| a. | GFC506 |
| b. | GFC507 |
| c. | GFC508 |
| d. | None of the above |