### Question 1

Answer saved

Marked out of 2.00

Flag question

#### Question text

Which of the following are examples of an aggregation relationship? Select all that apply.

Select one or more:

classroom and desks

parent and child

animal and dogs

sky and clouds

school and students

### Question 2

Answer saved

Marked out of 2.00

Flag question

#### Question text

A static method can do which of the following? Select all that apply.

Select one or more:

reference static variables in the class

reference instance data variables in the class

invoke non-static methods in the class

invoke other static methods in the class

### Question 3

Answer saved

Marked out of 2.00

Flag question

#### Question text

A non-static (instance) method can do which of the following? Select all that apply.

Select one or more:

invoke static methods in the class

reference instance data variables in the class

reference static variables in the class

invoke other non-static methods in the class

### Question 4

Answer saved

Marked out of 2.00

Flag question

#### Question text

Based on the following code fragments, which of the following are legal statements? Select all that apply.

public class Bird {

public Bird() {

...

}

public int getFlightSpeed() { ...}

public static int getMaximumFlightSpeed() { ...}

}

Select one or more:

int n = Bird.getFlightSpeed();

Bird b = new Bird();  
int n = b.getFlightSpeed();

Bird b = new Bird();  
b.getFlightSpeed();

int n = Bird.getMaximumFlightSpeed();

### Question 5

Answer saved

Marked out of 2.00

Flag question

#### Question text

What is printed when the main method below is run?

public class MyClass {

int num;

public MyClass() {

num = 5;

}

public void doSomething(int num) {

System.out.print(this.num);

System.out.print(num);

}

}

public static void main(String[] args) {

MyClass myClass = new MyClass();

myClass.doSomething(3);

}

Select one:

35

53

an error will be thrown

none of the above

### Question 6

Answer saved

Marked out of 6.00

Flag question

#### Question text

Explain, in your own words, the two uses of the "this" keyword.

The first use of "this" keyword would be for a field. This method is used very frequently for fields since fields are shadowed by a method or constructor parameter.

The second use of "this" keyword would be for a constructor. Inside a constructor, a person can use the "this" keyword to call another constructor in the same class.

### Question 7

Answer saved

Marked out of 4.00

Flag question

#### Question text

What is printed by the code fragment below?   
  
Rectangle r = new Rectangle(10, 10);   
int n = 4;   
shrink(r);   
shrink( n );   
System.out.print(r.getWidth() + "" + r.getHeight() + "" + n);   
  
public void shrink(Rectangle rec) {

int newWidth = (int) rec.getWidth()/2;  
int newHeight = (int) rec.getHeight()/2;  
rec .setSize(newWidth, newHeight);

}   
public void shrink(int q) {

q = q / 2;

}

Answer: 

### Question 8

Answer saved

Marked out of 4.00

Flag question

#### Question text

What is printed by the code fragment below?   
  
Rectangle r = new Rectangle(10, 10);   
int n = 4;   
shrink(r);   
shrink( n );   
System.out.print(r.getWidth() + "" + r.getHeight() + "" + n);   
  
public void shrink(Rectangle rec) {

rec = new Rectangle(2,2);  
int newWidth = (int) rec.getWidth()/2;  
int newHeight = (int) rec.getHeight()/2;  
rec.setSize(newWidth, newHeight);

}   
public void shrink(int q) {

q = q / 2;

}

Answer: 

### Question 9

Answer saved

Marked out of 2.00

Flag question

#### Question text

A class can contain both of the following method definitions:

public int howMany (int numberOfCouples) {

return 2 \* numberOfCouples;

}   
public double howMany (int numberOfCouples) {

return 2.3 \* numberOfCouples;

}

Select one:

True

False

### Question 10

Answer saved

Marked out of 2.00

Flag question

#### Question text

A class can contain both of the following method definitions:

public int howMany () {

return ( (int) howMany(1));

}   
public double howMany (int num) {

return 2.3 \* num;

}

Select one:

True

False

### Question 11

Answer saved

Marked out of 2.00

Flag question

#### Question text

A class can contain both of the following method definitions:

public int convertedValue(int number) {

if(number > 0)

return number;

return 0;

}   
public int convertedValue(int numberToConvert) {

if(number > 0)

return number;

return 0;

}

Select one:

True

False

### Question 12

Answer saved

Marked out of 2.00

Flag question

#### Question text

What happens when this is used in a constructor’s body to call another constructor of the same class if that call is not the first statement in the constructor?

Select one:

a compilation error occurs

a runtime error occurs

nothing happens- the program compiles and runs

a logic error occurs

**Question 13**

Not yet answered

Marked out of 14.00

Flag question

**Question text**

Write three overloaded methods that calculate averages. The first method takes two integer parameters, the second takes three integer parameters, and the third takes four integer parameters. Each method returns a double. Make sure all methods are calculating the correct average and that the three methods are properly overloaded.

public double averageFirst(int num1, int num2){

int sum;

double average;

sum = num1 + num2;

average = sum / 2;

}

public double averageSecond(int num1, int num2, int num3){

int sum;

double average;

sum = num1 + num2 + num3;

average = sum / 3;

}

public double averageThird(int num1, int num2, int num3, int num4){

int sum;

double average;

sum = num1 + num2 + num3 + num4;

average = sum / 4;

}

### Question 14

Not yet answered

Marked out of 34.00

Flag question

#### Question text

Write a class called Customer. A customer is described by:

* a name
* an ArrayList of Invoice objects

Your class should contain:

* two constructors
  + one constructor sets up a Customer with only a name (this means they currently have no invoices due)
  + the second constructor sets up a Customer with a name an one initial Invoice
* getters and setters
* a toString method; the text representation should include the customer name and the number of invoices the customer has
* a getTotalDue method that returns the total amount due by the customer (the sum of all invoices)
* an addInvoice method that adds an invoice for the current customer
* a counter to keep track of how many **customer objects** have been created

The Invoice class is given below. Use good principles of class design.

public class Invoice {

private String invoiceID;

private double amountDue;

public Invoice(String invoiceID, double amountDue) {

this.invoiceID = invoiceID;

this.amountDue = amountDue;

}

public String getInvoiceID() {

return invoiceID;

}

public double getAmountDue() {

return amountDue;

}

}