Q1. Which integer between 1 and 10000 has the largest number of divisors, and how many divisors does it have? Write a program to find the answers and print out the results. It is possible that several integers in this range have the same, maximum number of divisors. Your program only has to print out one of them (which occurs first).

Q2. If you have N Banana, then you have N/12 dozen’s of Bannana , with N%12 Banana’s left over. Write a program that asks the user how many Banana’s he/she has and then tells the user how many dozen’s of Banana’s she/he has in terms of Gross, Dozens and LeftOver.

**Questions and Programs on Static keyword.**

public class Student {

static int counter = 0;

int id;

public Student() {

id = counter;

counter++;

}

public static void main(String[] args)

{

Student s1 = new Student();

Student s2 = new Student();

Student s3 = new Student();

System.out.println(s1.id + "" + s2.id + "" + s3.id);

System.out.println(s1.counter + "" + s2.counter + ""+ s3.counter);

}

}

Q.1) Which is the result printed on screen of executing:

System.out.println(s1.id + "" + s2.id + "" + s3.id);

**ANS: 012**

System.out.println(s1.counter + "" + s2.counter + "" + s3.counter);

**ANS:** **333**

**EXPLANATION**

Class atributes (i.e. those declared as static have the same value for all the instances of that class, while instance attributes can have different values for each instance of that class.

**Questions and Programs on Inheritance and Super Keyword.**

public class MyClass

{

private int a;

public MyClass (int a)

{

this.a = a;

}

}

public class YourClass extends MyClass

{

private int b;

public YourClass (int a, int b) {

//...[Here]...

this.b = b;

}

}

**Given MyClass and YourClass classes, as shown below, what line should replace //...[Here]... inYourClass?**

**Ans:** super(a);

**Explanation:**

The initialization of attributes of a parent class from a child class requires the use of the keyword super, which invokes the constructor of the parent class with the corresponding arguments.

**Questions and Programs on Constructor Overloading.**

Which of the following sentences about constructors in Java are correct?

1. The return type void must be included in the declaration of a constructor.
2. A class can have several constructors.
3. Constructors are used to create objects (instances) of a class.
4. Constructors need to receive at least one argument.
5. Constructors are special methods that cannot be overloaded.

**Ans:** 2 and 3

Top of Form

**Explanation:**

Constructors are used to create objects of a class and always have the same name of the class. Their structure is the keyword public plus the name of the class plus a number of parameters (0, 1, 2...). Constructors do not include a return type, since they will return an object of the class. We can have multiple constructors in a class. Actually it is quite common to overload the constructor of a class to include several possibilities in the creation of objects (with one argument, with two arguments, etc.).

1. What is the o/p of the following Program?

class A {

public int i;

private int j;

}

class B extends A {

void display() {

super.j = super.i + 1;

System.out.println(super.i + " " + super.j);

} }

class inheritance {

public static void main(String args[])

{ B obj = new B();

obj.i=1;

obj.j=2;

obj.display();

}

}

**Ans: error(j has private access in A)**



class A {

public int i;

public int j;

A()

{

i = 1;

j = 2;

}

}

class B extends A {

int a;

B() {

super();

}

}

class super\_use {

public static void main(String args[])

{

B obj = new B();

System.out.println(obj.i + " " + obj.j) ;

}

}

Ans:1 2

Q)

class A {

public int i;

protected int j;

}

class B extends A {

void display() {

super.j = 3;

System.out.println(i + " " + j);

}

}

class Output {

public static void main(String args[])

{

B obj = new B();

obj.i=1;

obj.j=2;

obj.display();

}

}

Ans: 1 3

public class JavaApplication37 {

static int i=2;

public static void main(String[] args) {

System.out.println(" "+i);

m(i);

System.out.println(" "+i);

}

static public void m(int i)

{

i=i+2;

}

}

Ans: 2 2