Constructor in Square Class

```
import java.util.* ;
import java.io.*;
class Square {

    // Write your code here
    int length;
    Square()
    {
        length = 10;
    }
    Square(int l)
    {
        length = l;
    }
    public void printArea()
    {
        System.out.println(length*length);
    }
}

class Solution {

    public static void main(String args[]) {

        // Write your code here
        Square obj1 = new Square();
        obj1.printArea();
        Square obj2 = new Square(7);
        obj2.printArea();
    }
}
```

Which of the following statements about constructors is/are correct?

Options:

- ○ The constructor name should be the same as the class name.
- ○ The default constructor invokes super() and sets all instance variables to a default value such as 0, null.
- ○ If we do not define a constructor for a class, the compiler will generate one for us
- ○ If we want to call parent class constructor, we must call it in the constructor's first line.
- ◉ All the above

✓ Correct Answer

What is the return type of constructor?

Options:

- ○ void
- ○ int
- ◉ The constructor does not have a return type
- ○ None of the above

✓ Correct Answer

What is the output of the following code?

```
public class Solution {

    public Solution() {
        System.out.println("Default constructor called");
    }

    public Solution(int x) {
        System.out.println("Parameterized constructor called");
    }

    public static void main(String args[]) {
        Solution obj = new Solution();
    }

}
```

Options:

- ○ 0
- ◉ Default constructor called
- ○ Parameterized constructor called
- ○ None of the above

✔ Correct Answer

What is the output of the following code?

```
public class Solution {

    public Solution() {
        System.out.println("Default constructor called");
    }

    public static void main(String args[]) {
        Solution obj = new Solution(10);
    }

}
```

Options:

- ○ Default constructor called
- ○ 10
- ◉ Compile-time error
- ○ None of the above

✔ Correct Answer

Is it possible to overload a constructor by changing its return type?

Options:

- ○ Yes
- ◉ No

✔ Correct Answer