1.

Consider this program, which intends to print the word delimited by the characters '*' in its both ends:

```
public class SearchString {
    public static void main(String[] args) {
        String quote = "An *onion* a day keeps everyone away!";
        // match the word delimited by *'s
        int startDelimit = quote.indexOf('*');
        int endDelimit = quote.lastIndexOf("*");
        System.out.println(quote.substring(startDelimit, endDelimit));
    }
}
```

This program will print one of the following options:

A. *onion
B. *onion*
C. onion
D. *onio e. nion*

2.

Consider the following program:

```
class UsePrintf {
    public static void main(String[] args) {
        int c = 'a';
        float f = 10;
        long ell = 100L;
        System.out.printf("char val is %c, float val is %f, long int val is
                          %ld \n", c, f, ell);
    }
}
```

Which one of the following options best describes the behavior of this program when executed?

A. The program prints the following: char val is a, float val is 10.000000, long int val is 100.
B. The program prints the following: char val is 65, float val is 10.000000, long int val is 100.
C. The program prints the following: char val is a, float val is 10, long int val is 100L.
D. The program prints the following: char val is 65, float val is 10.000000, long int val is 100L.
E. The program prints the following: char val is 65, float val is 10, long int val is 100L.
F. The program throws an exception of java.util.UnknownFormatConversionException: Conversion = 'l'.

3.

Consider the following program:

```
class StrEqual {
    public static void main(String[] args) {
        String s1 = "hi";
        String s2 = new String("hi");
        String s3 = "hi";
        if (s1 == s2) {
            System.out.println("s1 and s2 equal");
        } else {
            System.out.println("s1 and s2 not equal");
        }
        if (s1 == s3) {
            System.out.println("s1 and s3 equal");
        } else {
            System.out.println("s1 and s3 not equal");
        }
    }
}
```

Which one of the following options provides the output of this program when executed?

A. s1 and s2 equal
   s1 and s3 equal

B. s1 and s2 equal
   s1 and s3 not equal

C. s1 and s2 not equal s1 and s3 equal

D. s1 and s2 not equal s1 and s3 not equal

4.

What will be the output of this program?

```
class Color {
    int red, green, blue;

    void Color() {
        red = 10;
        green = 10;
        blue = 10;
    }

    void printColor() {
        System.out.println("red: " + red + " green: " + green + " blue: " +
                           blue);
    }

    public static void main(String[] args) {
        Color color = new Color();
        color.printColor();
    }
}
```

A. Compiler error: no constructor provided for the class.
B. Compiles without errors, and when run, it prints the following: red: 0 green: 0 blue: 0.
C. Compiles without errors, and when run, it prints the following: red: 10 green: 10 blue: 10.
D. Compiles without errors, and when run, crashes by throwing NullPointerException.

5.

Look at the following code and predict the output:

```
class Color {
    int red, green, blue;

    Color() {
        Color(10, 10, 10);
    }

    Color(int r, int g, int b) {
        red = r;
        green = g;
        blue = b;
    }

    void printColor() {
        System.out.println("red: " + red + " green: " + green + " blue: "
                           + blue);
    }

    public static void main(String[] args) {
        Color color = new Color();
        color.printColor();
    }
}
```

A. Compiler error: cannot find symbol.

B. Compiles without errors, and when run, it prints the following: red: 0 green: 0 blue: 0.

C. Compiles without errors, and when run, it prints the following: red: 10 green: 10 blue: 10.

D. Compiles without errors, and when run, crashes by throwing NullPointerException.

6.

Choose the best option based on the following program:

```
class Color {
    int red, green, blue;

    Color() {
        this(10, 10, 10);
    }

    Color(int r, int g, int b) {
        red = r;
        green = g;
        blue = b;
    }

    public String toString() {
        return "The color is: " + red + green + blue;
    }

    public static void main(String[] args) {
        // implicitly invoke toString method
        System.out.println(new Color());
    }
}
```

A. Compiler error: incompatible types.

B. Compiles without errors, and when run, it prints the following: the color is: 30.

C. Compiles without errors, and when run, it prints the following: the color is: 101010.

D. Compiles without errors, and when run, it prints the following: the color is: red green blue.

7.

Choose the best option based on the following program:

```
class Color {
    int red, green, blue;

    Color() {
        this(10, 10, 10);
    }

    Color(int r, int g, int b) {
        red = r;
        green = g;
        blue = b;
    }

    String toString() {
        return "The color is: " + " red = " + red + " green = " + green
                + " blue = " + blue;
    }

    public static void main(String[] args) {
        // implicitly invoke toString method
        System.out.println(new Color());
    }
}
```

A. Compiler error: attempting to assign weaker access privileges; toString was public in Object.

B. Compiles without errors, and when run, it prints the following: the color is: red = 10 green = 10 blue = 10.

C. Compiles without errors, and when run, it prints the following: the color is: red = 0 green = 0 blue = 0.

D. Compiles without errors, and when run, it throws ClassCastException.

Answer

1. A

The substring (beginIndex, endIndex) consists of characters beginning from the character at beginIndex till the character at the endIndex – 1.

2. F

3. C

4. B

Remember that a constructor does not have a return type; if a return type is provided, it is treated as a method in that class. in this case, since Color had void return type, it became a method named Color() in the Color class, with the default Color constructor provided by the compiler. By default, data values are initialized to zero, hence the output.

5. A

The compiler looks for the method Color() when it reaches this statement: Color(10, 10, 10);. The right way to call another constructor is to use the this keyword as follows: this(10, 10, 10);.

6. C

The toString() implementation has the expression “the color is: “ + red + blue + green. Since the first entry is string, the + operation becomes the string concatenation operator with resulting string “the color is: 10”. Following that, again there is a concatenation operator + and so on until finally it prints “the color is: 101010”.

7. A

No access modifier is specified for the toString() method. Object's toString() method has a public access modifier; you cannot reduce the visibility of the method. hence, it will result in a compiler error.