Question1:

```
class Phone {
    void call() {
        System.out.println("Call-Phone");
    }
}
class SmartPhone extends Phone{
    void call() {
        System.out.println("Call-SmartPhone");
    }
}
class TestPhones {
    public static void main(String[] args) {
        Phone phone = new Phone();
        Phone smartPhone = new SmartPhone();
        phone.call();
        smartPhone.call();
    }
}
```

a. Call-Phone
   Call-Phone

b. Call-Phone
   Call-SmartPhone

c. Call-Phone
   null

d. null
   Call-SmartPhone

Question2:

```
class Phone {
    String keyboard = "in-built";
}
class Tablet extends Phone {
    boolean playMovie = false;
}
class College2 {
    public static void main(String args[]) {
        Phone phone = new Tablet();
        System.out.println(phone.keyboard + ":" + phone.playMovie);
    }
}
```

a. in-built:false

b. in-built:true

c. null:false

d. null:true

e. Compilation error

Question 3:

```
public class If2 {
    public static void main(String args[]) {
        int a = 10; int b = 20; boolean c = false;
        if (b > a) if (++a == 10) if (c!=true) System.out.println(1);
        else System.out.println(2); else System.out.println(3);
    }
}
```

- a 1
- b 2
- c 3
- d No output

Question 4:

```
class Course {
    int enrollments;
}
class TestEJavaCourse {
    public static void main(String args[]) {
        Course c1 = new Course();
        Course c2 = new Course();
        c1.enrollments = 100;
        c2.enrollments = 200;
        System.out.println(c1.enrollments + c2.enrollments);
    }
}
```

What will happen if the variable `enrollments` is defined as a `static` variable? (Select 1 option.)

- a No change in output. `TestEJavaCourse` prints `300`.
- b Change in output. `TestEJavaCourse` prints `200`.
- c Change in output. `TestEJavaCourse` prints `400`.
- d The class `TestEJavaCourse` fails to compile.

Question 5:

```
class EMyMethods {
    static String name = "m1";
    void riverRafting() {
        String name = "m2";
        if (8 > 2) {
            String name = "m3";
            System.out.println(name);
        }
    }
    public static void main(String[] args) {
        EMyMethods m1 = new EMyMethods();
        m1.riverRafting();
    }
}
```

- a m1
- b m2
- c m3
- d The code fails to compile.

Question 6:

```
class Bottle {
    void Bottle() {}
    void Bottle(WaterBottle w) {}
}
class WaterBottle extends Bottle {}
```

- a A base class can't pass reference variables of its defined class as method parameters in constructors.
- b The class compiles successfully—a base class can use reference variables of its derived class as method parameters.
- c The class `Bottle` defines two overloaded constructors.
- d The class `Bottle` can access only one constructor.

Question 7:

```
class Book {
    private int pages = 100;
}
class Magazine extends Book {
    private int interviews = 2;
    private int totalPages() { /* INSERT CODE HERE */ }

    public static void main(String[] args) {
        System.out.println(new Magazine().totalPages());
    }

}
```

- a `return super.pages + this.interviews*5;`
- b `return this.pages + this.interviews*5;`
- c `return super.pages + interviews*5;`

Question 8:

```
Byte b1 = (byte)100;                        // 1
Integer i1 = (int)200;                      // 2
Long l1 = (long)300;                        // 3
Float f1 = (float)b1 + (
     0int)l1;              // 4
String s1 = 300;                            // 5
if (s1 == (b1 + i1))                        // 6
    s1 = (String)500;                       // 7
else                                        // 8
    f1 = (int)100;                          // 9
System.out.println(s1 + ":" + f1);          // 10
```

what is the output? Select 1 option.

- a Code fails compilation at line numbers 1, 3, 4, 7.
- b Code fails compilation at line numbers 6, 7.
- c Code fails compilation at line numbers 7, 9.
- d Code fails compilation at line numbers 4, 5, 6, 7, 9.
- e No compilation error—outputs `500:300`.
- f No compilation error—outputs `300:100`.
- g Runtime exception.

Question 9:

```
class EIf {
    public static void main(String args[]) {
        bool boolean = false;
        do {
            if (boolean = true)
                System.out.println("true");
            else
                System.out.println("false");
        }
        while(3.3 + 4.7 > 8);    }
}
```

- a The class will print `true`.
- b The class will print `false`.
- c The class will print `true` if the `if` condition is changed to `boolean == true`.
- d The class will print `false` if the `if` condition is changed to `boolean != true`.
- e The class won't compile.
- f Runtime exception.

Question 10:

```
class Book {
    String ISBN;
    Book(String val) {
        ISBN = val;
    }
}
class TestEquals {
    public static void main(String... args) {
        Book b1 = new Book("1234-4657");
        Book b2 = new Book("1234-4657");
        System.out.print(b1.equals(b2) +":");
        System.out.print(b1 == b2);
    }
}
```

- a `true:false`
- b `true:true`
- c `false:true`
- d `false:false`
- e Compilation error—there is no `equals` method in the class `Book`.
- f Runtime exception.