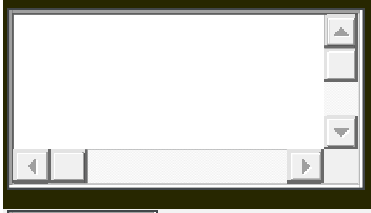
1.

Question 1

Suppose you have the code below. Is ‘int printSomething(int i, int j) { ... }’ a valid overloaded method?

```
public void printSomething(int i) {
    System.out.print(i);
}
```



1 / 1 point

True

False

Correct

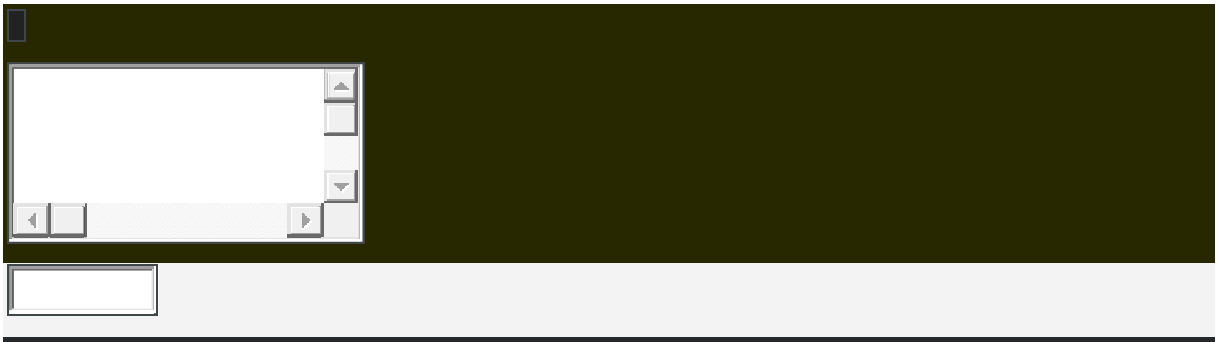
It is a valid overloaded method because the number of arguments for printSomething is different. The method also happens to have a different return type.

2.

Question 2

Suppose you have the code below. Is ‘void printSomething(String i) { … }’ a valid overloaded method?

```
public void printSomething(int i) {
    System.out.print(i);
}
```



1 / 1 point

TRUE

FALSE

Correct
It is a valid overloaded method because the type of the argument for printSomething is different.

3.

Question 3

You can have two overloaded methods with a difference only in the return type.

1 / 1 point

True

False

Correct
To overload methods, you have to have a difference either in name or argument types or number of arguments.

4.

Question 4

A method signature consists of:

1 / 1 point

Method Name, Return Type and Number of Arguments

Method Name, Number of Arguments, Types of Arguments, and Order of Arguments

Access Modifier, Method Name and Types of Arguments

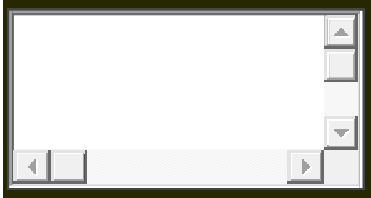
Correct
A method signature consists of the method name and the number of arguments, types of arguments, and order of arguments.

5.

Question 5

Suppose you have the code below. Is 'String printSomething(String i) { … }' a valid overloaded method?

```
public void printSomething(int i) {
    System.out.print(i);
}
```



1 / 1 point

True

False

Correct
It is a valid overloaded method because the type of the argument for printSomething is different. The method also happens to have a different return type.

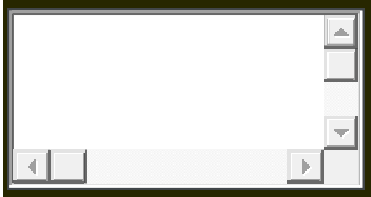
6.

Question 6

Suppose you have the code below. Is 'void printSomething(char i, int j) { ... }' a valid overloaded method?

4

```
public void printSomething(int i, char j) {
    System.out.print(i + ", " + j);
}
```



1 / 1 point

- True
- False

Correct
It is a valid overloaded method because the order of arguments for printSomething is different.