

COMP 1409 Introduction to Software Development 1 Quiz #4 Sample Solution

1. The following method will not compile. Explain what is wrong and rewrite it so that it compiles and does what its comment says it will do. Assume that accountBalance is a double field in a Customer class. (2 points)

```
/**
 * Returns the balance of the account and sets balance to zero
 * @return account balance
 */
public double refundAccountBalance()
{
    return accountBalance;
    accountBalance = 0.0;
}
```

There cannot be an executable statement after a return. If we want to both return the amount in accountBalance and set accountBalance to zero we must use a local variable to temporarily store the value in accountBalance.

```
public double refundAccountBalance()
{
    double tempBalance = accountBalance;
    accountBalance = 0.0;
    return tempBalance;
}
```

2. The following method will not compile. Explain what is wrong and rewrite it. Assume that accountPaid is a boolean field in a Customer class. (2 points)

```
/**
 * @return true if the customer account is paid or false if not
 */
public boolean isGoodCustomer()
{
    if(accountPaid == true) {
        return true;
    }
}
```

This method must return a boolean, either true or false.

```
public boolean isGoodCustomer()
{
    if(accountPaid == true) {
```

```

        return true;
    }
    else {
        return false;
    }
}

```

Alternatively:

```

public boolean isGoodCustomer()
{
    if(accountPaid) {
        return true;
    }
    return false;
}

```

Or:

```

public boolean isGoodCustomer()
{
    return accountPaid; // it's already either true or false!
}

```

3. For each of the following segments of code, write what the final value of x will be.
(4 points total: 1 point each)

(a) `int x = 6;`
`int y = 7;`
`x = x + y;`

x == 13

(b) `boolean x = false;`

`x = !x;`
`x = x || x;`

x == true

(c) `double x = 25.5;`
`double y = 25.4;`
`if(y <= x) {`
 `x = y;`
`}`

x == 25.4

```
(d)  int y = 6;
      boolean x = ((y >= 6) && (y == 0))

      x == false
```

4. Describe clearly when a compound-AND statement can short circuit, i.e. not require the second part to be evaluated. **(1 point)**

AND is true only when both parts of the statement are true. Therefore if the first part is false the second part does not need to be evaluated as the statement is automatically false.

5. Fill in the following method named setYearOfBirth with the following behavior. **(3 points total)**

- if the parameter's value is between 1900 and 2012 inclusive, the parameter's value is assigned to an instance variable named yearOfBirth **(1 point)**
- if the parameter's value is less than 1900, do not assign its value to yearOfBirth; instead, write "too early" to the screen **(1 point)**
- if the parameter's value is more than 2012, do not assign its value to yearOfBirth; instead, write "too late" to the screen. **(1 point)**

```
public void setYearOfBirth(int newYearOfBirth)
{
    if(newYearOfBirth < 1900) {
        System.out.println("too early");
    }
    else if(newYearOfBirth > 2012) {
        System.out.println("too late");
    }
    else {
        yearOfBirth = newYearOfBirth;
    }
}
```