Name: Solutions

Chapter 10 Quiz

1. [10pts] Name 5 things you should add and/or update when planning the next iteration.

① Add/update user stories
② Update velocity
③ Update burn-down graph
④ Update big board
⑤ Update estimates
⑥ Update priorities
⑦ Update tasks
⑧ Update task assignments

2. [2pts] If you have pending bugs, how do you prioritize which bugs to fix in the next iteration?

Ask the customer — customers set priorities!

3. [2pts] T or F. New third-party code should be treated with great suspicion.

T

4. [12pts] Circle and annotate 6 different problems you can see in the third-party code below. You're looking for everything from readability of the code to problems with functionality. Note that in testing a program that used this code, it was found that the program would hang when submitting orders, giving no other indications of a problem. (The lines are double spaced to make them easier for you to annotate.)

```
// Follows the Singleton design pattern
```

unhelpful comment

```
public class MercuryMeals

{

     public MercuryMeals meallythang;

     private Order cO;

     private String qk = "select * from order-table where keywords
like %1;";
```

bad names (unhelpful)

bad indentation

```
public MercuryMeals() {

}
```

no need to explicitly define

```
public MercuryMeals getInstance() {

     this.meallythang = new MercuryMeals();

     return this.instance;

}
```

Violates singleton pattern

pending to-do

```
// TODO Really should document this at some point... TBD

public Order createOrder() {

     return new Order();

}
```

seems pointless

Continued next page...

No doc

```
public MealOption getMealOption(String option) throws
MercuryMealsConnectionException {

    if (MM.establish().isAnyOptionsForKey(option)) {

        return MM.establish.getMealOption(option).[0]

    }

    return null;

}

public boolean submitOrder(Order cO) {

    try {

        MM mm = MM.establish();

        mm.su(this.cO);

    } catch (Exception e) {

        // write out an error message

    }

    return false;

}

public Order[] getOrdersThatMatchKeyword(String qk) throws
MercuryMealsConnectionException {

    Order o = new Order[];

    try {

        o = MM.establish().find(qk, qk);

    } catch (Exception e) { return null; }

    return o;

}}
```

()

;

why establish twice?

exception caught, but not really handled

qk shadowing confusing; looks like a bug

bad indentation