

5.3.5. 🌸 Programming Challenge : Comments and Conditions

Working in pairs or groups, come up with 4 steps that a user must do to purchase a product, for example a book on Java, in an online store, and list the preconditions and postconditions for each step. You could pretend to buy something online to come up with the steps. (You could use an online drawing tool like Creately.com (choose Use-Case Diagrams) to draw a Use-Case diagram for the Online Store System, but it is not required). Don't forget to list the preconditions and postconditions for each step. You can type in your answer below.

5-3-8: Write down 4 steps that a user must do to purchase a product, for example a book on Java, in an online store, and list the preconditions and postconditions for each step.

- sign in with username and password
- search for a product
- choose a specific product
- compare with similar products
- decide which to take based on price and value
- enter banking information
- complete transaction
- receive item

Save

Instructor's Feedback

Your answer has been saved.

Activity: 5.3.5.1 shortanswer (challenge-5-3-use-case-preconditions)

Here is a simple class called User that could be used in an online store. Add good commenting to this code before the class, the instance variables, and the methods.

Save & Run



Original - 1 of 1

Download

Show CodeLens

Reformat



Pair?

```
1 //Code by Akshat Garg
2 public class User
3 {
4     //holds user information private from other users
5     private String username;
6     private String password;
7
8     //default user constructor used for guest accounts
9     public User()
10    {
11        username = "guest";
12        password = "guest" + (int) (Math.random() * 1000);
13    }
14
15    //user-entry based username & password
16    public User(String nameInit, String pwordInit)
17    {
18        username = nameInit;
19        password = pwordInit;
20    }
21
22    //displays welcome message
23    public void welcome()
24    {
25        System.out.println("Welcome " + username + "!");
26    }
27
28    //creates two users, one guest and one login, and welcomes both
29    public static void main(String[] args)
30    {
31        User u1 = new User(); // guest login
32        // new user
33        User u2 = new User("cooldude@gmail.com", "Coolness*10");
34        u1.welcome();
35        u2.welcome();
36    }
37 }
38
39 --
```

Activity: 5.3.5.2 ActiveCode (challenge-5-3-comments)

Create a Pet class that keeps track of the name, age, weight, type of animal, and breed for records at an animal clinic.

Save & Run

7/17/2024, 1:16:22 PM - 4 of 4

Download

Show CodeLens

Reformat

Pair?

```
1/**
2 * Pet class (complete comments)
3 *
4 * Akshat Gang
5 * 7/17/24
6 */
7class Pet
8{
9    // keep track of the name, age, weight, type of animal, and breed of the pet
10   private String name;
11   private int age;
12   private double weight;
13   private String type;
14   private String breed;
15   // Write a constructor, accessor (get) methods, and a toString method. Use go
16   // commenting.
17
18   //creates default pet
19   public Pet(){
20       name = "Fido";
21       age = 5;
22       weight = 10.0;
23       type = "dog";
24       breed = "terrier";
25   }
26   //user entered info pet constructor
27   public Pet(String initName, int initAge, double initWeight,
28               String initType, String initBreed){
29       name = initName;
30       age = initAge;
31       weight = initWeight;
32       type = initType;
33       breed = initBreed;
34   }
35   //returns attributes of pet
36   public String getName(){
37       return name;
38   }
39   public int getAge(){
40       return age;
41   }
42 }
```

Fido, age 5, weighs 10.0 pounds and is a terrier dog
Socks, age 3, weighs 15.0 pounds and is a calico cat
Your cat weighs 15.0

Result	Expected	Actual	Notes
Pass	3+ line(s) of text	3 line(s) of text	Checking main method prints info for 3 Pet objects
Pass	2+	2	Checking for constructor
Pass	5	5	Checking accessor (get) methods for each variable
Pass	true	true	Checking that code contains toString() method
Pass	2	2	Checking main method creates 2 Pet objects
Pass	5 Private	5 Private	Checking Private Instance Variables

You got 6 out of 6 correct. 100.00%

Activity: 5.4.2.1 ActiveCode (challenge-5-4-Pet-Class)

Create a Pet class that keeps track of the name, age, weight, type of animal, and breed for records at an animal clinic.

Save & Run

7/17/2024, 1:16:22 PM - 4 of 4

Download

Show CodeLens

Reformat

Pair?

```
38 }
39 public int getAge(){
40     return age;
41 }
42 public double getWeight(){
43     return weight;
44 }
45 public String getType(){
46     return type;
47 }
48 public String getBreed(){
49     return breed;
50 }
51 //returns all data in string format
52 public String toString(){
53     return name + ", age " + age + ", weighs " + weight + " pounds and is a "
54         + breed + " " + type;
55 }
56 // Don't forget to complete the main method in the TesterClass below!
57
58
59 public class TesterClass
60 {
61     // main method for testing
62     public static void main(String[] args)
63     {
64         // Create 2 Pet objects and test all your methods
65         Pet doggy = new Pet();
66         System.out.println(doggy.toString());
67         Pet kitty = new Pet("Socks", 3, 15, "cat", "calico");
68         System.out.println(kitty.toString());
69         System.out.println("Your " + kitty.getType() + " weighs "
70             + kitty.getWeight());
71     }
72 }
73
74 }
```

Fido, age 5, weighs 10.0 pounds and is a terrier dog
Socks, age 3, weighs 15.0 pounds and is a calico cat
Your cat weighs 15.0

Result	Expected	Actual	Notes
Pass	3+ line(s) of text	3 line(s) of text	Checking main method prints info for 3 Pet objects
Pass	2+	2	Checking for constructor
Pass	5	5	Checking accessor (get) methods for each variable
Pass	true	true	Checking that code contains toString() method
Pass	2	2	Checking main method creates 2 Pet objects
Pass	5 Private	5 Private	Checking Private Instance Variables

You got 6 out of 6 correct. 100.00%

Activity: 5.4.2.1 ActiveCode (challenge-5-4-Pet-Class)

Fun larger exercise in making a nice coherent running program. Learned a lot in keeping comments to differentiate different sections of work