Humna Sultan
Mr. Hendricks
AP CS A
18 September 2020

1. **Exercise R3.1:** Why is the BankAccount(double initialBalance) constructor not strictly necessary?
    a. Two constructors are not necessary and a constructor without parameters is acceptable.
2. **Exercise R3.2:** Explain the difference between BankAccount b; and BankAccount b = new BankAccount(5000);
    a. The first one creates an object of BankAccount, while the second one initializes an object using a constructor.
3. **Exercise R3.3:** Explain the difference between new BankAccount(5000); and BankAccount b = new BankAccount(5000);
    a. The first one creates an object of the class BankAccount, while the second one initializes an object of the BankAccount class using a constructor.
4. **Exercise R3.4:** What happens in our implementation of the BankAccount class when more money is withdrawn from the account than the current balance?
    a. The bank balance becomes negative.
5. **Exercise R3.5:** What is the value of b.getBalance() after the following operations? BankAccount b = new BankAccount(10); b.deposit(5000); b.withdraw(b.getBalance() / 2);
    a. (10+5000)/2 = $2505; 5100-2505 = $2595
6. **Exercise R3.6:** If b1 and b2 refer to objects of class BankAccount, consider the following instructions.
    a. b1.deposit(b2.getBalance());
    b. b2.deposit(b1.getBalance());
        i. Are the balances of b1 and b2 now identical? Explain.
            1. The balances are not identical; there are two separate objects with different parameters. Because they are different objects, they can have different method calls, parameters, values, and variables.
7. **Exercise R3.7:** What is the *this* reference? Why would you use it?
    a. The *this* reference is used to refer to the object of the current class. You would use it to avoid naming conflicts between method or constructor of the object.
8. **Exercise R3.8:** What does the following method do? Give an example of how you can call the method.

public class BankAccount
{

public void mystery(BankAccount that,
double amount)
{
this.balance = this.balance - amount;
that.balance = that.balance + amount;
}
. . . // Other bank account methods
}

The purpose of the method is to transfer money from one account to another by subtracting an amount from one account and adding to another. The way you could call the method would be b1.mystery(b2,100);

9. **Exercise R3.10:** What are the accessors and mutators of the CashRegister class?
    a. Accessors:
        i. Access Specifier: public class cashRegister(){}
        ii. Return Type: return change;
        iii. Method Name: recordPurchase, enterPayment, giveChange
        iv. List of Parameters: double amount
        v. Method body contained in brackets
    b. Mutators:
        i. recordPurchase method
10. **Exercise R3.11:** Explain the difference between a local variable and a parameter variable.
    a. A local variable is a variable that only exists in a specific method and is declared/initialized within that method, while a parameter variable is passed to a specific method from another for use.
11. **Exercise R3.12:** Explain the difference between an instance field and a local variable.
    a. An instance field is the field of an instance of a class or object, while a local variable is a variable declared/initialized within a method to only be used in that method.

Exercise P3.4. : Implement a class Employee. An employee has a name (a string) and a salary (a double). Provide a constructor with two parameters

public Employee(String employeeName, double currentSalary) and methods

public String getName()
public double getSalary()
public void raiseSalary(double byPercent)

These methods return the name and salary, and raise the employee's salary by a certain percentage. Sample usage:
Employee harry = new Employee("Hacker, Harry", 50000);
harry.raiseSalary(10); // Harry gets a 10% raise

Supply an EmployeeTester class that tests all methods.

Exercise P3.6. :
Implement a class Student. For the purpose of this exercise, a student has a name and a total quiz score.

Supply an appropriate constructor and methods:
- *getName()*
- *addQuiz(int score)*
- *getTotalScore()*
- *getAverageScore()*

To compute the latter, you also need to store the number of quizzes that the student took.

Supply a StudentTester class that tests all methods.