



New York University  
Computer Science Department  
Introduction to Software Development  
Dr. Anasse Bari

**Homework#4: Learning Methods in Java**

**Deadline:** See NYUclasses for the deadline, 15% off per day after the deadline (3 days maximum).

**Learning Objective:**

- Practicing Methods and Modularization in Java to solve real world problems

**Read the guidelines bellow carefully to avoid receiving a zero grade on the HW:**

- Please use the following naming convention "ExerciseX.java", where X is the number of the exercise. Please apply this convention to all coming homework assignments during the semester. Also please attach the answers and include them into HW's zip file.
- Compile and run the program for each exercises.
- **Comment Your Code (at least 5 comments per exercise)**
- Make an archive (zip file or compressed file) with all the **java files (the .java files NOT the .class files)** and post it on NYU Classes (StudentName\_Homework\_X.zip)
- It is your responsibility to make sure that the Zip files has your actual files. You may send the file to yourself by email to double check that is the actual file before you upload on NYU classes.
- **If the graders cannot open the file, you will receive a grade of zero.**
- **If you send the .class files instead of the .java files (source files) you will receive a zero.**
- Please **do not send the entire project**, the Java file for each exercise is all that we need.
- An act of cheating will be severely addressed with an immediate zero on the homework and a report to the academic advisor and the administration.
- You will automatically lose 50% of the points for an exercise if the program does not compile and run correctly.
- The number of points for each questions is being specified.

- Plagiarized assignment will get a ZERO grade. You cannot change the variable names of other student's solution and submit it as yours. The program structure of other students must not match yours. Every student has to come up with his/her own solution. Any cheating (e.g. copying from internet without citing sources) is a serious violation of the University student code.

### Programming Problem 1 (20 points)

#### ATM Machine *with Methods*

Consider the same problem from the previous HW.

Design a simple ATM service kiosk. This kiosk supports the following options for its menu:

1. View your account balance
2. Deposit Cash
3. Withdraw cash
4. Exit

Initially, the account balance is \$0.00. When you deposit cash, this account balance is increased by your deposit amount. Similarly, when you withdraw cash, this account balance is decreased by your withdrawal amount. You cannot withdraw more than what exists in the account. Similarly, you cannot deposit a negative cash amount. Use if-else statements or switch statement to perform the respective operation in the menu. The ATM kiosk provides these three options until the user selects the option to leave the kiosk. At the end of every transaction, the user is being asked if they need to perform another transaction. You may want to use a **do-while statement** to satisfy this last requirement.

You will to provide a solution to this problem using Modularization. You will need to define and implement the necessary methods to implement an ATM.

Most of the methods that you will define needs to have inputs and outputs.

You cannot use more than two methods with no inputs and/or outputs.

Example and expected output:

```
1. View your Balance
2. Deposit Cash
3. Withdraw Cash
4. Exit

Enter your selection: 1

Your current balance is 0.00
1. View your Balance
2. Deposit Cash
3. Withdraw Cash
4. Exit

Enter your selection: 2

Enter the amount you want to deposit: 100
1. View your Balance
2. Deposit Cash
3. Withdraw Cash
4. Exit

Enter your selection: 3

Enter the amount your want to withdraw: 97
1. View your Balance
2. Deposit Cash
3. Withdraw Cash
4. Exit

Enter your selection: 1

Your current balance is 3.00
1. View your Balance
2. Deposit Cash
3. Withdraw Cash
4. Exit

Enter your selection: 3

Enter the amount your want to withdraw: 5
Sorry you don't have enough balance!
1. View your Balance
2. Deposit Cash
3. Withdraw Cash
4. Exit

Enter your selection: 4

Goodbye
```

Please note that for these exercise you will need to write the methods and call them in your program (for each exercise you will need to write a complete Java program where you call your methods)

**From all the exercises bellow, you will need to write the Java method and write a complete Java program that calls each method in each exercise.**

- **Exercise 2 (10pts)** Write a method in Java that takes two integers and returns the last digit of the sum of the two numbers (you should **not** use the math library for this exercise)  
e.g. if the two number are 49 and 2  
then the method should return 1  
(the last digit of 51 is 1)
- **Exercise 3 (10pts)** Write a method that calculates an area of a circle given a radius as an input. Call the method you define in the main and test your problem. The input is being given by the user in the main and sent to the method as an input.
- **Exercise 4 (10pts)** Write a method in Java that takes a number as an input and returns a 1 if the number is even and 0 if the number is odd. The input is being given by the user in the main and sent to the method as an input.
- **Exercise 5 (10pts)** Write a method in Java that takes two numbers (ints) as input and determines if the second is a multiple of the first. The inputs are being given by the user in the main and sent to the method as an input.
- **Exercise 6 (10pts)** Write a method that returns and generates randomly a letter in the alphabet. The method needs to be called in the main.
- **Exercise 7 (10pts)** Write a method that two numbers that represent a range (a,b), the methods returns and generates randomly an integer within a given range. The method needs to be called in the main.
- **Exercise 8 (10pts)** Write a Java method that takes a number and prints it in reverse order.

That is, if the whole received 123, should print the integer 321. (you are NOT allowed to use arrays)

**You are not allowed to use arrays, strings or Java's math library. The input to the method is an int.**

- **Exercise 9 (10pts)** Write program in java that accepts users input and validates if the input is a palindrome. A palindrome is a word that reads the same forwards as it does backwards. For this exercise we will consider a palindrome a palindrome without punctuation.

Examples of palindromes:

My gym

Top spot