

Programming Exercise 01 - Basic Java

Authors: Vince, Akul, Andrew

Problem Description

This assignment will test your basic knowledge of variables, expressions, assignment, classes, and Strings.

You will solve 2 problems:

1. Calculating your change after taxes.
2. Outputting someone's birthday.

Solution Description

Change after buying groceries

1. Create a class called `Groceries`
2. Add to this class a main method.
3. In the main method do the following:
4. Declare a variable called `name` and assign to it the name `Brad`.
5. Declare a variable called `cash` and assign to it the value `50`. Make this variable an `int` because cash comes in whole numbers.
6. Declare a variable called `taxRate` and assign to it the value `0.1`. This variable should be the larger of the two primitive types that can hold floating point numbers.
7. Declare a variable called `subtotal` and assign to it the value `42.3`. This variable should be the larger of the two primitive types that can hold floating point numbers.
8. Calculate the change after taxes and store it in a variable called `change`. Use the same type as `taxRate`. Perform this calculation using a single expression.
9. Use a single `printf` statement to print the following lines to the console:

```
{name} has ${change} dollars remaining!  
  
{name} started with ${cash} dollars!
```

where `{name}` is the value of `name` declared above, `{change}` is the value of `change` that was calculated, to two decimal places, and `{cash}` is the amount of money we started with. Make sure to use `System.out.printf` here, not `System.out.println` or `System.out.print`.

10. For the specified values of `name`, `cash`, `taxRate`, and `subtotal` the output should be

```
Brad has $3.47 dollars remaining!  
  
Brad started with $50 dollars!
```

NOTE: the dollar amount for change shown should be displayed to two decimal places.

Magic birthday calculator

1. Create a class called `BirthdayCalculator`
2. Add to this class a main method that does the following:
 1. Declare an `int` variable called `birthMonth` and assign to it `1`.

- II. Declare an int variable called `birthDay` and assign to it 23.
- III. Declare an int variable called `birthYear` and assign to it 1996.
- IV. Declare an int variable called `bdayCalculation` and assign it equal to `birthMonth` (Don't just put 1 here, use `birthMonth`).
- V. Do the following operations to `bdayCalculation`. Use the Compound Assignment operators (e.g. `*=`, `+=`, etc.).
 - i. Add 18.
 - ii. Multiply by 25.
 - iii. Subtract 333.
 - iv. Multiply by 8.
 - v. Subtract 554.
 - vi. Divide by 2.
 - vii. Add `birthDay`.
 - viii. Multiply by 5.
 - ix. Add 692.
 - x. Multiply by 20.
- VI. Add the last two digits of `birthYear` (use the modulus operator to get the last two digits).
- VII. Subtract 32940.
- VIII. Print out to the console `Birthday: {bdayCalculation}` where `{bdayCalculation}` is the birthday that was calculated.
- IX. For the specified values of `birthMonth`, `birthDay`, and `birthYear`, the output should be `Birthday: 012396`.
(Side note: JDK1.0 was released on January 23, 1996.)

Testing

Income after taxes

Try changing the values for `name`, `cash`, `taxRate`, and `subtotal` and verify that the output is still correct.

Magic age calculator

Try changing the values for `birthMonth`, `birthDay`, and `birthYear` and verify that the output is still correct. `birthYear` must be four digits.

Import Restrictions

You may not import anything for this homework assignment.

Feature Restrictions

There are a few features and methods in Java that overly simplify the concepts we are trying to teach or break our auto grader. For that reason, do not use any of the following in your final submission:

- `var` (the reserved keyword)
- `System.exit`

Collaboration

Collaboration Statement

To ensure that you acknowledge collaboration and give credit where credit is due, **we require that you place a collaboration statement as a comment at the top of at least one java file that you submit**. That collaboration statement should say either:

I worked on the homework assignment alone, using only course materials.

or

In order to help learn course concepts, I worked on the homework with [give the names of the people you worked with], discussed homework topics and issues with [provide names of people], and/or consulted related material that can be found at [cite any other materials not provided as course materials for CS 1331 that assisted your learning].

Recall that comments are special lines in Java that begin with `//`.

Turn-In Procedure

Submission

To submit, upload the files listed below to the corresponding assignment on Gradescope:

- `Groceries.java`
- `BirthdayCalculator.java`

Make sure you see the message stating "PE01 submitted successfully". From this point, Gradescope will run a basic autograder on your submission as discussed in the next section.

You can submit as many times as you want before the deadline, so feel free to resubmit as you make substantial progress on the homework. We will only grade your last submission: be sure to **submit every file each time you resubmit**.

Gradescope Autograder

For each submission, you will be able to see the results of a few basic test cases on your code. If you fail a test, you can look at the output to determine what went wrong and resubmit once you have fixed the issue.

The Gradescope tests serve two main purposes:

1. Prevent upload mistakes (e.g. forgetting checkstyle, non-compiling code)
2. Provide basic formatting and usage validation

In other words, the test cases on Gradescope are by no means comprehensive. Be sure to thoroughly test your code by considering edge cases and writing your own test files. You also should avoid using Gradescope to compile, run, or checkstyle your code; you can do that locally on your machine.

Other portions of your assignment can also be graded by a TA once the submission deadline has passed, so the output on Gradescope may not necessarily reflect your grade for the assignment.

Important Notes (Don't Skip)

- Non-compiling files will receive a 0 for all associated rubric items
- Do not submit `.class` files.
- Test your code in addition to the basic checks on Gradescope
- Submit every file each time you resubmit
- Read the "Allowed Imports" and "Restricted Features" to avoid losing points
- Check on Piazza for all official clarifications