

Problem Description

The `GregorianCalendar` class describes a point in time, as measured by the Gregorian calendar, the standard calendar that is commonly used throughout the world today. You construct a `GregorianCalendar` object from a year, month, and day of the month. This example constructs `GregorianCalendar` objects for today and for the birthday of J. Presper Eckert.¹

```
GregorianCalendar today = new GregorianCalendar();
GregorianCalendar eckertsBirthday = new GregorianCalendar(1919,
    Calendar.APRIL, 9);
```

Use the constants `Calendar.JANUARY`, ..., `Calendar.DECEMBER` to specify the month.

If you want to add a number of days to a `GregorianCalendar` object, use the `add` method:

```
today.add(Calendar.DAY_OF_MONTH, 10);
GregorianCalendar tenDaysFromNow = today;
```

This is a *mutator* method — it changes the `today` object. Since the name `today` is no longer appropriate, we created a new object with a descriptive name. *You should do the same in your program.* When a name is no longer appropriate, use a new one.

The `get` method can be used to query a given `GregorianCalendar` object:

```
int month = today.get(Calendar.MONTH) + 1;
int day = today.get(Calendar.DAY_OF_MONTH);
int year = today.get(Calendar.YEAR);
int weekday = myBirthday.get(Calendar.DAY_OF_WEEK);
```

Note that for the weekday, 1 is Sunday, 2 is Monday, ..., 7 is Saturday. For the month, 0 is January, 1 is February, ..., 11 is December.

Your task is to write a class `MyCalendar` that prints the following information:

- The current date;
- The weekday of your birthday;
- The date that is 10,000 days from your birthday;
- The date and weekday for October 4, 1582. The date and weekday for the next day. (Compute the next day by using the `add` method to add one.)

¹J. Presper Eckert was one of the designers of ENIAC. See page 7 of your text.

The result may surprise you. Read the introduction to `GregorianCalendar` in the Java API <<http://java.sun.com/javase/6/docs/api/>> to find out why. Insert a comment in your program that explains the result.

Details

Here is a check list that you should review before submitting your project. These details are important, so please check the list carefully.

- Name your class `MyCalendar`.
- When you create your project in Eclipse, call it `proj1-xyz01` where `xyz01` is your LINUX login name.
- Make sure that your name and section number are clearly identified in comments at the top of your program.
- When you use the `get` method to determine a day of the week, it will return a number representing the day as explained above. You do *not* need to translate this into a string with the name of the day. It's not too hard to do though. If you decide to do it, use the method `getDisplayName`.
- Here is an example showing how the first part of your output must look.

```
Today is 9/19/2010
The weekday of my birthday is Wednesday.
10,000 days from my birthday is 8/6/1978
```

- Your class name *must* begin with an uppercase letter.
- Don't forget to add one to the month.
- Remove all warnings before submitting your project. Eclipse will also underline spelling errors in your comments. Fix those too!
- Variable names *must* begin with a lowercase letter. Use camel casing. Use descriptive name such as `myBirthday`, *not* `date1`.
- If Eclipse is not reformatting your code every time you save, please ask for help.
- Surround comments with blank lines so they are easier to read.

- After adding 10,000 to your birthday, use a new variable name.
- If you are using your own computer, make sure the Eclipse settings are the same as lab 3.

What to Submit

Drag your [proj1-xyz01](#) folder in the the drop box with your instructor's name, *not* the lab drop box.