Franklin Marathon Test-Driven Development Plan

TDD Requirement 1.1:
Given a date stamp in the format of YYYYMMDD, the system shall be able to determine the location of that time within the race calendar year, according to the following table:

Dates	Race Calendar Period
1 Jun – 30 Sep	Registration Not Open
Oct 1 – Oct 31	Super Early
Nov 1 – Feb 28/29	Early
Mar 1 – Apr 1	Baseline
Apr 2 – TDay	Late
TDay – 31 May	Registration Closed

where TDay is the Thursday before the first Saturday of May.

TDD Requirement 1.2:
Given a date of birth, the system shall be able to calculate the age of a runner on race day. (Race day is TDay + 2 for the 5K and 10K, and TDay + 3 for the full and half marathons.)

Iterations 1 & 2 (HW3a) are due on the first Tuesday after this project is assigned

TDD Requirement 2.1:
Given the information in the column on the left, the system shall add a runner to the race roster that includes the information in the column on the right.

Info from Registration	Info Stored in Race Roster
a. First Name	a. First Name
b. Last Name	b. Last Name
c. Date of Birth	c. Age on Race Day
d. Gender	d. Gender
e. Email Address	e. Email Address
f. Registration Timestamp	f. Registration Timestamp

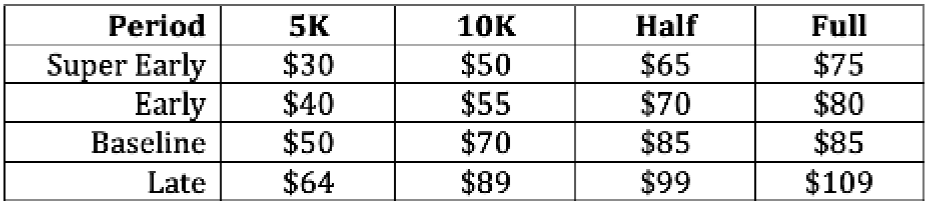
(At this point in the development effort, there should only be one race roster.)

TDD Requirement 2.2:
The system shall allow a runner to sign up for any one of four races (5K, 10K, Half Marathon, and Full Marathon) and add the runner's information to the correct roster.

(At this point in the development effort, there should be four race rosters – one for each race. A Race Distance may need to be added to the registration data, but it does not need to be included in each data record.)

Iterations 3 & 4 (HW3b) are due on the second Tuesday after this project is assigned

TDD Requirement 3.1:
Given a registration timestamp and a race distance, the system shall calculate the correct price for the race using the following table:



Period	5K	10K	Half	Full
Super Early	$30	$50	$65	$75
Early	$40	$55	$70	$80
Baseline	$50	$70	$85	$85
Late	$64	$89	$99	$109

TDD Requirement 3.2:
The system shall add an “Amount Paid” field to the race roster records showing how much money each runner paid in entry fees, to include a $5 discount for any runners age 65 or older on the day they registered for the race.

Iterations 5 & 6 (HW3c) are due on the third Tuesday after this project is assigned

TDD Requirement 4.1:
The system shall allow a runner to sign up for two races (a Saturday race and a Sunday race) and add a 20% discount to sum of the two races. When this happens, the runner’s race roster information will be added to both rosters.

TDD Requirement 4.2:
The system will allow no more than 100 runners to sign up for events on Saturday, and the system will allow no more than 100 runners to sign up for events on Sunday.

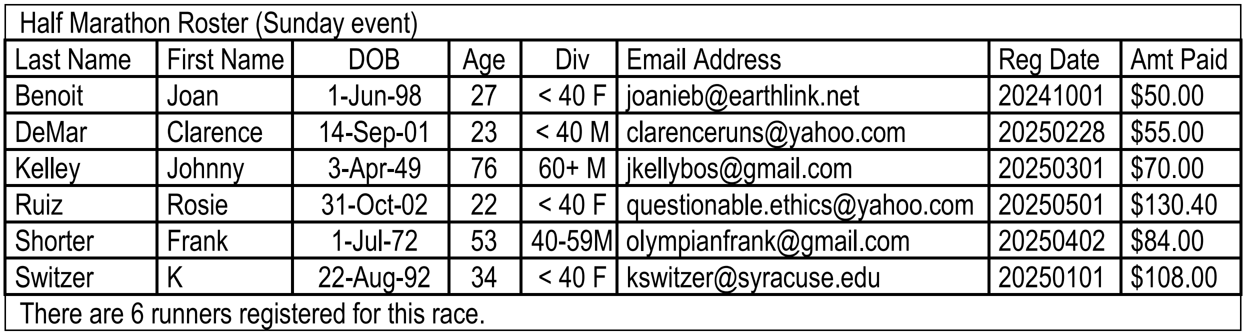
Iterations 7 & 8 (HW3d) are due on the fourth Tuesday after this project is assigned

TDD Requirement 5.1:
The system shall be able to print off rosters for all races, showing these fields for each runner:

Info Stored in Race Roster
a. Last Name
b. First Name
c. Date of Birth
d. Age on Race Day
e. Division
f. Email Address
g. Registration Date & Amt Paid

At the bottom of each roster, the system will print the number of runners who have signed up for that event. (See the example table on the next page to see what this table should look like.)

Iteration 9 (Proj 3) is due on the fifth Tuesday after this project is assigned
Note: Project 3 should also include a full acceptance test plan that should be run and annotated as part of Project 3.



Half Marathon Roster (Sunday event)

Last Name	First Name	DOB	Age	Div	Email Address	Reg Date	Amt Paid
Benoit	Joan	1-Jun-98	27	< 40 F	joanieb@earthlink.net	20241001	$50.00
DeMar	Clarence	14-Sep-01	23	< 40 M	clarenceruns@yahoo.com	20250228	$55.00
Kelley	Johnny	3-Apr-49	76	60+ M	jkellybos@gmail.com	20250301	$70.00
Ruiz	Rosie	31-Oct-02	22	< 40 F	questionable.ethics@yahoo.com	20250501	$130.40
Shorter	Frank	1-Jul-72	53	40-59M	olympianfrank@gmail.com	20250402	$84.00
Switzer	K	22-Aug-92	34	< 40 F	kswitzer@syracuse.edu	20250101	$108.00

There are 6 runners registered for this race.

Additional Notes:

1. There are six age divisions (three for each gender): under 40, 40-59, and 60 and older. (Actual races typically have several more divisions, but we are trying to keep things simple.)

2. For this assignment, **always** use test-driven development (TDD) to build your code. This means you must start by writing test cases specific to each requirement, and then writing just enough code to pass those tests.

3. At the end of each iteration, run the test cases you wrote for that iteration, and be sure to document the actual results.

4. Since most weeks have two iterations, your submission for that week should be structured in a [test plan, test plan, code, code] format, as follows:
    a. List the first requirement followed by your test cases for the first iteration. Assuming you were able to complete the coding in time, include the annotated actual results, to include whether or not the test passed or failed.
    b. List the second requirement followed by your test cases for the second iteration, also annotated with actual results.
    c. Provide the code for the first iteration, followed by the code for the second iteration. These should **<u>not</u>** be combined as one! That said, if a later iteration has code that gets folded into code written for prior iterations, that's fine, but you should still turn in two snippets of code to illustrate how the system is gradually growing during the TDD process. Remember to take a "snapshot" of your code from the first iteration before modifying that code in a subsequent iteration.

5. You *might* be able to reuse some of your test cases from Project 2, but you should not panic if those don't fit. Because we have shifted to lower levels of the V-Model, lower-level test cases written for unit testing may not align perfectly with higher-level test cases writing for acceptance testing.

6. You may need to refactor the code developed in earlier iterations as the software begins to be integrated. As discussed in class, that's normal in TDD.

7. The end state of your working software should be:

   - The software will calculate the runner’s race day age from DOB
   - The software can accept registrations and build rosters
   - The software will calculate registration fees and bundling/age discounts
   - The software will cap rosters at 100 runners per day
   - The software will display/print race day rosters

   That said, do not work ahead without sticking to the TDD plan outlined here.

8. As you work through this project, pay particular attention to how the TDD methodology has certain benefits, such as more reliable code with fewer bugs, especially the absence of hard-to-find bugs that often infest systems that are built in much larger chunks.

9. The sample roster shown on the previous page is sorted by last name. You can assume that the race staff can sort the table in Excel after retrieving it from the system. In other words, you can keep the original tables sorted in the order in which the registrations were received.

10. The alignment of requirements and iterations found in this document supersedes any conflicting prior guidance that may have been given verbally in class.

11. Do not wait until the last week to write your acceptance test plan. Work on it when you have some “down” time.