VICTORIA UNIVERSITY
OF WELLINGTON
Te Whare Wananga o te Upoko
o te Ika a Maui

School of Information Management (SIM)
Te Kura Whakaipurangi Korero

Assignment 1

Trimester 2 2011

Personal Fitness Tracker

<u>Due Date</u>: **Friday 12 August 5pm**
Assignment Demo – during Week 5

Overview

In this assignment, you must write a program which will help individuals to assess their fitness level and track calories burnt when exercising. The program will calculate the Body Mass Index (BMI) based on weight and height of the program user. It will also calculate the amount of burnt calories when performing various activities. Finally, it will count the number of simulations that are run by users.

During Week 5, you will have to make a short demo of your assignment to your tutors. They will ask you general questions about the assignment but they may also request you to explain particular pieces of code.

Please note:

- *Read the whole assignment before you begin.*
- *When you receive this assignment, you will find that you may not be familiar with some of the programming concepts required to create this application. However, as the lectures and workshop sessions progress, the main concepts will be covered. Therefore, it is very important that you attend the lectures and workshops as well as working in your own time to complete this assignment. In addition, some problems in this assignment require you to investigate the answers for yourself using the text book or other resources such as the C# help files.*

Learning Objectives

In this assignment you will be showing an understanding of the following:

- C# built-in classes, objects, and simple methods
- Interface design principles
- Interface controls: form, button, group box, text box, rich text box, message box, check box
- Variables, constants and calculations

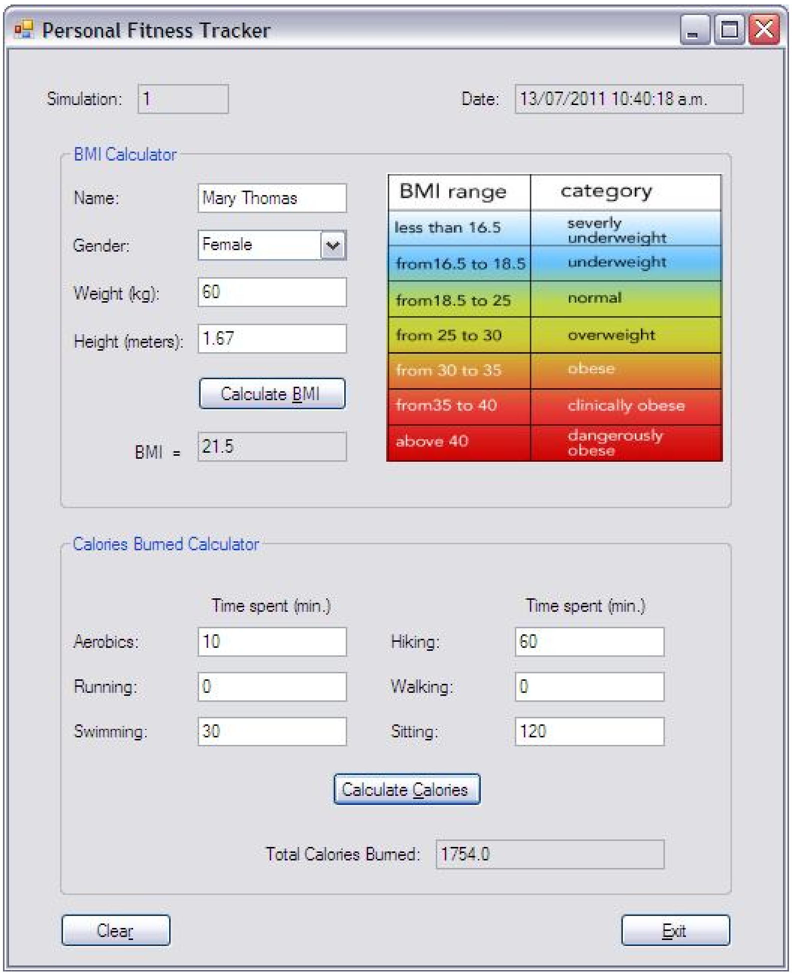
Warnings

1. You must write your program in C# 2008. Programs written in C# 2005 or C# 2010 will not be marked.
2. You are responsible for backing up your own work.
3. Plagiarism is not accepted in this course. Please read the course outline for guidelines on the University plagiarism policy.

Instructions - Personal Fitness Tracker

- You are required to create a C# application that could form part of a computer system for individuals who are interested in monitoring their fitness level.
- Your application (program) will allow the end user to enter his/her weight (in kg) and height (in meters) and to generate his/her BMI by clicking on a button.
- Your application will allow the end user to enter the amount of time (in minutes) spent performing various activities and the amount of burnt calories will be generated when clicking on a button.
- The program will tally the total number of simulations that are run by the end user.
- The program will generate the current date and time each time the program is started by the end user.
- You will create a single form to capture the data from the end user.
- The end user may make a mistake when entering data. Ensure that all extreme and unexpected data are handled (ie. data is present and all data is reasonable).

The form is shown below:



Personal Fitness Tracker

Simulation: 1

Date: 13/07/2011 10:40:18 a.m.

BMI Calculator

Name: Mary Thomas

Gender: Female

Weight (kg): 60

Height (meters): 1.67

BMI = 21.5

BMI range	category
less than 16.5	severly underweight
from16.5 to 18.5	underweight
from18.5 to 25	normal
from 25 to 30	overweight
from 30 to 35	obese
from35 to 40	clinically obese
above 40	dangerously obese

Calories Burned Calculator

	Time spent (min.)		Time spent (min.)
Aerobics:	10	Hiking:	60
Running:	0	Walking:	0
Swimming:	30	Sitting:	120

Total Calories Burned: 1754.0

Requirements are as follows:

1	All program objects including variables, constants, and methods are named following good programming standards: code layout, naming convention, comments (as specified in the text book).
2	The application will have a single form exactly as shown in figure 1 which will have two group boxes to organize the data entry areas (group boxes: BMI and Calories calculations)
3	When the form opens the current date and time will be displayed. Use `DateTime.Now`. You will need to convert this value into a string.
4	The form will allow the following data to be entered: a. The name of the end user b. The gender of the end user. Populate the combo box using the Items property. Do not do this using C# code. c. The weight (in kg) and the height (in meters) of the end user. d. The time spent performing various activities: aerobics, running, swimming, hiking, walking, and sitting.
5	The form will display a BMI chart in a picture box in the top right hand corner of the form. The image file is provided.
6	The form will have appropriate ToolTips on editable textboxes and buttons.
7	The tab order will be set appropriately.
8	Details for the form and data entry controls are as follows a. Name the form *FitnessTrackerForm*, set the StartPosition property to CenterScreen and the ControlBox property to false. b. When a text box is used to display data only (and not to enter data), its Read-Only property should be TRUE making the text box appeared greyed-out. c. All textboxes should have a fixed3D border style. d. Comment the code appropriately, as you go. e. Remove empty events from the code. ***Important remark***: You are free to modify the colour scheme and overall appearance of the program as long as the application looks professional. Keep in mind the principles of GUI design seen in class. However, the location of the controls cannot be modified.
9	The form will have four buttons; each button will have an access key as shown in the figure on page 3. The buttons are a. *A Calculate BMI* button to calculate and display the BMI of the end user using the following formula: $\textbf{\textit{BMI = weight in kilograms / height in meters}}^2$ ***Important remark***: *Declare the BMI, weight, and height variables as **decimal**. Round the value of the BMI to 2 decimal numbers by using the method **Math.Round** (see Help in Visual Studio).*

b. A *Calculate calories* button to calculate the total amount of calories burnt while performing various physical activities. Please use the following values to compute the number of calories burnt for each type of activity.

Activity	**calories/min.**
Aerobics	12.5
Running	14.2
Sitting	1.3
Walking	5.7
Hiking	18.9
Swimming	11.3

Important remark: *Declare all variables related to the calculation of calories as* ***decimal****. Round the total number of burnt calories to 1 decimal number by using the method* ***Math.Round*** *(see Help in Visual Studio).*

Example: if a person spends 30 minutes hiking, the number of burnt calories is:
30 × 18.9 = 567 calories (the calculation does not take into consideration the weight of the end user).

c. A *Clear* button to clear all values in the form except the date/time and the number of simulations.

Important remark: To clear the combo box, use:

```
nameComboBox.SelectedIndex = -1;
```

d. An *Exit* button to close the form and end the program.

10 Use a class level constant for the number of simulations. Keep a count of the number of simulations run entered into the program (each time an end user calculates BMI or number of burnt calories).

Each time the end user clicks on either the *Calculate BMI* or *the Calculate calories* button, the number of simulations to be displayed must be increased by one.

11 You are expected to handle the exceptions that are triggered when a user does not enter numerical values.

12 You are expected to *add an <u>extra feature</u> of your choice.*

You are free to add any feature which you think would improve the application (it could be adding a control such as a button or a text box, including the code necessary to handle related events).

A strategy for proceeding:

1. Draw the form on a piece of paper. Draw all the controls and name them (following naming convention seen in class). For each control, write in pseudo-code both properties and associated methods.
2. Set up the form in Visual Studio, rename the form (do not use the default name of Form1), set the properties for the form (text, centered). Do not set the control box to false yet.
3. Add controls to the form, name the controls appropriately following good naming standards and set the properties for each control.
4. Write code for the *Exit* button.
5. Set the Form Control Box property to false.
6. Set up the Tooltips. You only need ONE Tooltip control for the whole form.
7. Set the Tab order.
8. Write code for the *Clear* button.
9. Write the form load event; get the date and time to display.
10. Write the code to count the number of simulations.
11. Write the code for the BMI calculation. Get this to work before proceeding.
12. Write the code for the calories calculation.

The calculations of the BMI and the number of calories are not related. It is recommended to fully implement the BMI calculation feature before dealing with the calculation related to the number of calories.

A good tip is to compile your program frequently and fix the problems that occur before you proceed from one step to the next (in general, it is recommended to compile and run the application each time you have done a change, even a minor one).

Notes and submission instructions

1. **The C# product.** You can use C# Visual Express 2008 or Visual C#.Net 2008 Professional edition to create this application. Both versions are suitable for this assignment. Instructions for downloading these products for home use are available on Blackboard. For downloading at home you will find the Express edition is smaller.
2. **Submitting your C# application**. You will need to zip your project folder and then use file transfer protocol (FTP) to submit the zipped folder to the SIM web server. There will be a practice session on these procedures in the timetabled workshop. Instructions for zipping project folders and for using FTP are available on Blackboard. Ensure you know how to do this because if you do not submit your assignment correctly we cannot mark it.

Indicative Marking Guide

GUIDELINES			**Mark**	**Student Mark**
Demo	1	Student is able to: • Explain the functionalities of the application. • Explain the overall structure of the program.	3	
	2	Student is able to make minor modifications in the code in response to tutor request.	2	
Project Assessment	3	Program executes on opening.	2	
	4	Form layout follows requirements • Controls (including correct location). • Image correctly displayed and hot keys are set. • Date displays correctly.	5	
	5	Calculate buttons work • Correct calculation of BMI. • Correct calculation of number of calories. • Number of simulations correctly increases.	5	
	6	Clear button works • All controls cleared except date/time and simulation number.	2	
	7	Exit button works and code is correct.	1	
	8	Code comments are present and appropriate • Main comment • Method comments	2	
	9	Naming standards correct and consistent throughout • Form and other controls • Variables • Constant	3	
	10	Clean code • Correct layout (indentation) • No empty events	2	
	11	Extra feature • The suggested extra feature is relevant. • The suggested extra feature provides value to the application.	3	
Total			**30**	

10% of total course mark