ITAP1001
Software Development Fundamentals

Assignment 1

July 2022

Weightage: 10%

Project Submission deadline: Session 9

Individual Assignment

The purpose of this assignment is to assess students on the following Learning Outcomes:

LO1: Explain the basic programming concepts such as computer storage and data types, decision structures, loops, and modular programming.
LO2: Use problem solving techniques and program design methods to translate a problem description to a formal program specification.
LO3: Differentiate various object-oriented programming constructs, including classes, inheritance, polymorphism, and encapsulation.

Details & Problems

In this first assignment, you are required to answer two short questions, give output of the code, and develop two C# Console Programs to demonstrate your ability to use C# input/output via command line, C# primitive and built-in C# types, C# operators and expression, C# conditional statements, C# loop construct, and show your ability to validate the inputs to avoid run-time errors.

Assignment Description

Object Oriented Programming (OOP) is a programming model where programs are organized around objects and data rather than action and logic. OOP allows decomposition of a problem into several entities called objects and then builds data and functions around these objects.

A class is the core of any modern Object-Oriented Programming language such as C#. In OOP languages it is mandatory to create a class for representing data.

A class is a blueprint of an object that contains variables for storing data and functions to perform operations on the data. A class will not occupy any memory space and hence it is only a logical representation of data.

You must have learnt it already in the lecture so far that to create a class, you simply use the keyword "class" followed by the class name. Also, some description about the Objects is provided below, which again you must have covered in the lectures.

Objects are the basic run-time entities of an object-oriented system. They may represent a person, a place, or any item that the program must handle. "An object is a software bundle of related variable and methods." "An object is an instance of a class".

All the programming languages supporting Object Oriented Programming will be supporting these three main concepts,

1. Encapsulation
2. Inheritance
3. Polymorphism

Abstraction provides you a generalized view of your classes or objects by providing relevant information.

By considering all this information provided above about the classes, objects, inheritance, polymorphism, and abstraction, answer the following questions:

Question:1. Suppose you have an object 'Mobile' phone and three models as provided below:

Nokia 1400 (Features: Calling, SMS)
Nokia 2700 (Features: Calling, SMS, FM Radio, MP3, Camera)
Black Berry (Features: Calling, SMS, FM Radio, MP3, Camera, Video Recording, Reading Emails)

Abstract information (necessary and common information) for the object "Mobile Phone" is that it makes a call to any number and can send SMS.

1. You need to write the code which will create abstract class for a Mobile phone object.

2. Write the formal program specification describing the names of the class, subclasses, and their attributes to be used for the above assignment description. Briefly describe the inheritance relationships between them.

Question:2. Write a program to accept the user's full name and email id. Validate the name and email id as per the following.

a. Name can only contain letters, spaces, and an apostrophe (').
b. Email should be in format <email>@<provider>
c. <email> can only have alphanumeric and "-" and "_" as special symbols and nothing else.
d. <email> can't start or end with a special symbol.

e. <provider> should have a minimum of 1 and maximum of 2 dot symbols, ".", in it and at least two letters on each side of the dot "." symbols.

Code related guidelines: Your C# console-based application should include the following:

1. The code should have a consistent, professional, and well-organized appearance.
2. Code should compile without errors.
3. Choose meaningful identifiers names.
4. The developed system takes inputs and handles the error in friendly manner.
5. Your code must adhere to OOPS concepts with implementation guidelines including modularity, reusability, extensibility, maintainability, and adaptability.

Marking Guide: 10 Marks

Task	Description	Marks
Console-based application code for Que. 1 Part A	Consistent and well written code with proper comments and no errors	3
Formal program specification for Que.1 Part B	Describing the relationships between different programming constructs	2
Validating operations	Well written executable code	5