

TITLE	HET: DIPLOMA IN INFORMATION TECHNOLOGY
SUBJECT	Programming 3B
SUBJECT CODE	PRG320
TEST/EXAM	Test Memo
SEMESTER	2nd
DATE WRITTEN	14 September 2021

NUMBER OF PAGES	11
TOTAL MARKS	100
DURATION	2 HOURS
PASS MARK	50%
WEIGHTING	20%
EXAMINER	Thandanani Dube

REQUIREMENTS:

Learner Requirements:

Equipment Requirement:

This paper consists of:

100 Marks

- | | |
|--------------------------------|----------|
| 1. Section A: Short Questions | 30 Marks |
| 2. Section B: Medium Questions | 10 Marks |
| 3. Section C: Long Questions | 60 Marks |

Unless otherwise stated, please answer ALL questions.

Please read the assessment rules and regulations that follow

Learners are warned that contravening any of the examination rules or disobeying the instructions of an invigilator could result in the examination being declared invalid. Disciplinary measures will be taken which may result in the students' expulsion from Damelin.

ASSESSMENT RULES AND REGULATIONS

Please ensure that you have read and fully understand the following assessment rules and regulations prior to commencing with your assessment:

1. To be permitted access to the examination, a learner must arrive with:
 - 1.1 an Identity Document or other official proof of identity (for example, - a student card, passport or driver's licence card with photo); and
 - 1.2 The required exam stationery.
2. No learner may enter the examination room more than 30 minutes after the examination sitting has commenced and no candidate may leave the room less than one hour after the examination sitting has commenced.
3. No extra time will be allowed should a student arrive late.
4. All learners must sign the *Attendance Register* for the examination on arrival.
5. It is the responsibility of learners to familiarise themselves with the examination rules prior to sitting for the examination.
6. All examinations are to be written on the date and time officially stipulated by the College.
7. It is the responsibility of learners to ensure that they are writing the correct paper and that the question paper is complete
8. Cell phones must be switched *off* prior to entering the exam venue. Cell phones and wallets may be placed under candidates' chairs rather than at the front of the room.
9. Learners may not handle cell phones or wallets during the exam.
10. No weapon of any description may be taken into the assessment room.
11. All personal belongings are to be placed at the front of the examination room. Personal belongings brought to the examination are at the owner's risk.
12. Smoking is not permitted and learners will not be allowed to leave the examination room in order to smoke
13. Once the examination has commenced, all conversation of any form between candidates must cease until after candidates have left the room, after the examination.
14. *Only* the official College examination book, as supplied by the College, may be used.
15. Learners must ensure that their student number is written on the answer book.
16. Learners are responsible for ensuring that they follow the instructions in the examination for submitting their answers.
17. Please read the instruction appearing on the examination paper carefully
18. The number of every question must be clearly indicated at the top of every answer.
19. No pages may be torn out of the answer book. All question papers and scrap paper must be handed to the invigilator after the examination.
20. Learners finishing earlier are to leave the examination room as quietly as possible on the instruction of the invigilator, and may not talk until outside the building where the examination is being written.
21. Only under exceptional circumstances will a learner be permitted to leave the examination room during the examination, and if the invigilator gives permission. An invigilator must accompany the learner. Only one learner at a time may be absent from the examination room.
22. Candidates may not act dishonestly in any respect.

SECTION A: SHORT QUESTIONS

[30]

Answer all questions in this section.

Question 1:

[20]

1.1 What are method parameters for which an argument is required in every method called?

- A. New parameters
- B. Class parameters
- C. **Mandatory parameters**
- D. Argument pass

1.2 An _____ to a method is one for which a default value is supplied automatically if you do not explicitly send one as an argument; a parameter is optional when it is given a value in the method declaration.

- A. Functional parameter
- B. Class parameter
- C. **Optional parameter**
- D. Object pass

1.3 A _____ in a method header receives a copy of the value passed to it.

- A. **Value parameter**
- B. Optional parameter
- C. Field parameter
- D. Object parameter

1.4 A _____ in a method header receives the parameter's address; a variable sent to a reference parameter is required to have been assigned a value before it is used in the method call.

- A. URI parameter
- B. **Reference parameter**

- C. File path parameter
- D. Browser parameter

1.5 An _____ in a method header receives the parameter's address; a variable sent to an output parameter is not required to have been assigned a value before it is used in the method call.

- A. Reference parameter
- B. **Output parameter**
- C. Input parameter
- D. Method call

1.6 A _____ is a local array declared within a method header that can accept any number of elements of the same data type.

- A. Object parameter
- B. File path parameter
- C. Input parameter
- D. **Parameter array**

1.7 An out _____ is a variable declared at the point where it is passed to a method as an out argument.

- A. Class
- B. Object
- C. Method
- D. **variable**

1.8 _____ involves the ability to write multiple versions of a method using the same method name but different parameter lists.

- A. **Overloading**
- B. Overriding
- C. Overlapping
- D. Class extends

1.9 A method's _____ is composed of its name and parameter list.

- A. Path
- B. **signature**
- C. Identifier
- D. Object

1.10 _____ is the process of determining which of multiple applicable methods is the best match for a method call.

- A. **Overload resolution**
- B. Method control
- C. Method override
- D. Underline resolution

Answer all questions in this section.

State whether true or false:

Question 2:

[10]

2.1 The TryParse () methods are an alternative to the Convert () and Parse ().

True

2.2 A parameter array is a local array declared within the method header by using the keyword

Array []

False, A parameter array is a local array declared within the method header by using the keyword **params.**

2.3 When you overload a C# method, you write multiple methods with Multiple names.

False, When you overload a C# method, you write multiple methods with a shared name.

2.4 The TryParse () method requires the receiving variable to be an out parameter for two reasons: The argument does not have an assigned value in the calling method. The method returns a Boolean value that indicates whether the conversion was successful and, so, it cannot return the score value too.

True

2.5 Methods are required to carry out tasks in an application. The Method describes the mechanisms that perform its task. In a class, you provide one or two methods designed to perform the class's tasks.

True

Define the following terms:

Question 3:

[10]

1. **Objects:** are reference types, and are created on the heap, using the keyword new.
2. **Access modifier:** determines which class methods -- including methods of other classes --can see and use a member variable or method within a class.
3. **Internal:** The members in class A that are marked internal are accessible to methods of any class in A's assembly.
4. **Private:** the members in class A that are marked private are accessible only to methods of class A
5. **Protected internal:** the members in class A that are marked protected internal are accessible to methods of class A, to methods of classes derived from class A and also to any class in A's assembly. This is effectively protected or internal (there is no concept of protected and internal)

Answer all questions in this section.

Practical question:

Question 4:

[30]

Declaring a class with a method and instantiating an Object of a class

The demonstration in this subtopic shall be using two classes; GradeBook and GradeBookTest. Class GradeBook will be used to display a message on the screen welcoming the instructor to the application. Class GradeBookTest is a testing class in which the Main method will create and use an object of the GradeBook class.

There are 10 errors in this demonstration fix them to allow the program to function.

```
// Class declaration with one method
using System;
public class GradeBook
{
//display a message to the grade book user
public void DisplayMessage()
{
Console.WriteLine("Welcome to the Grade Book!");
}
}
```

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
// create a GradeBook objects and call its DisplayMessage Method
public class GradeBookTest
{
public static void Main (String [] args)
{
```



```
//create a GradeBook objects and assign it to myGradeBook
GradeBook myGradeBook = new GradeBook ();
//call myGradeBook's Method
myGradeBook.DisplayMessage();
Console.Read();
}
}
```

```
using System;
// create a GradeBook objects and call its DisplayMessage Method
public class GradeBookTest
{
    public static void Main(String [] args)
    {
        //create a GradeBook objects and assign it to myGradeBook
        GradeBook myGradeBook = new GradeBook ();
        //Prompt for an input Coursename
        Console.WriteLine("Please enter the course name:");
        string nameofcourse = Console.ReadLine(); //read the line of text and store
        Console.WriteLine(); //output a blank space
        //call myGradeBook's Method
        //and pass the nameofcourse as an argument
        myGradeBook.DisplayMessage(nameofcourse);
        Console.Read();
    }
}
```

Question 5:

[30]

In C#, you create a derived class by adding a colon after the name of the derived class, followed by the name of the base class:

public class ListBox : Window

Create a code that declares a new class, ListBox, that derives from Window. You can read the colon as "derives from."

```
using System;
using System.Collections.Generic; using System.Linq; using System.Text;
using System.Threading.Tasks;
namespace Values
{
public class Window
{
// constructor takes two integers to
// fix location on the console
public Window(int top, int left)
{
this.top = top;
this.left = left;
}
// simulates drawing the window public void DrawWindow()
{
Console.WriteLine("Drawing Window at {0} , {1} ", top, left);
}
// these members are private and thus invisible // to derived class methods; we'll
examine this // later in the chapter private int top;
private int left;
// ListBox derives from Window
}
public class ListBox : Window{
```

```
// constructor adds a parameter
public ListBox(int top,int left, string theContents): base(top, left){ mListBoxContents
= theContents;
}
// a new version (note keyword) because in the
// derived method we change the behavior public new void DrawWindow( ){
base.DrawWindow( ); // invoke the base method
Console.WriteLine ("Writing string to the listbox: {0} ", mListBoxContents);
}
private string mListBoxContents;
}
public class Tester{
public static void Main( ) { // create a base instance
Window w = new Window(5,10); w.DrawWindow( );
// create a derived instance
ListBox lb = new ListBox(20,30,"Hello world"); lb.DrawWindow( );
}
}
}
```

TOTAL MARKS: 100