TITLE	HET: DIPLOMA IN INFORMATION TECHNOLOGY
SUBJECT	Programming 3B
SUBJECT CODE	PRG420
TEST/EXAM	Deferred Test Memo
SEMESTER	2nd
DATE WRITTEN	28 September 2021

NUMBER OF PAGES	10
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. URl 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 Inheritance is the principle that states you can apply your knowledge of a general category to more specific objects

True

2.2 When you create a class by making it inherit from another class, you are provided with data fields and methods automatically

True

2.3 A class that is used as a basis for inheritance is called a child class

False

2.4 When you create a class that inherits from a base class, it is a derived class or extended class

True

2.5 You can use the terms superclass and subclass as synonyms for base class and derived class

True

Define the following terms:

Question 3: [10]

1. **instance variables: of a class are the data components that exist separately for each instantiation**

2. **class header: or class definition describes a class; it contains an optional access modifier, the keyword class, and any legal identifier for the name of the class.**

3. **ref parameters: do not contain the values associated with the parameters, but they contain the memory addresses where the values are stored.**

4. **Out parameters: like ref parameters, contain memory addresses that are passed to a method, allowing them to alter the original variables**

5. **Parameter Arrays: A parameter array is a local array declared within the method header by using the keyword params.**

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!");
}
}
```

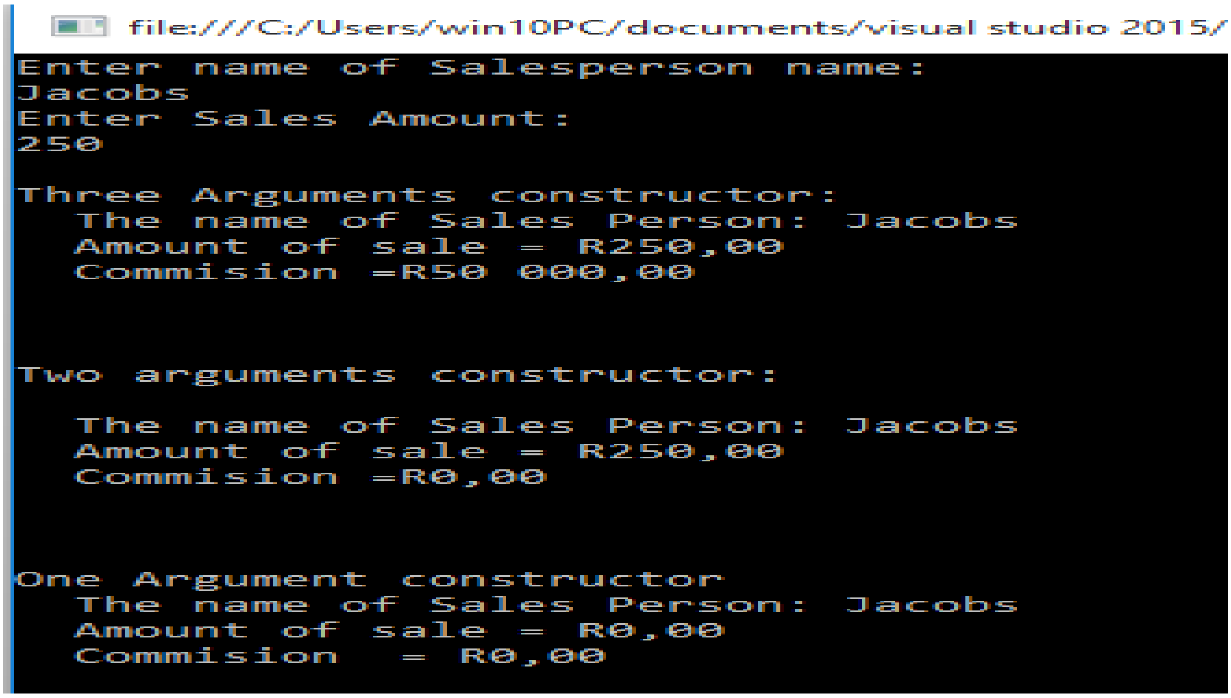
```
using System;
using System.Collections. Generic;
using System.Linq;
using System.Text;
// create a GradeBook objects and call its DisplayMessage Method
public class GradeBookTest
{
public static void Main (String [] args)
{
```

```
//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 nameofcource = 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(nameofcource);
Console.Read();
}
}
```

Question 5: **[30]**

Create a console application named "SalesTransactionDemo" that declares several SalesTransaction objects and displays their values and their sum. The SalesTransaction class contains fields for a salesperson name, sales amount, and commission and a read-only field that stores the commission rate which amount to R200 per sales amount. Include three constructors for the class. One constructor accepts values for the name, sales amount, and rate, and when the sales value is set, the constructor computes the commission as sales value times commission rate. The second constructor accepts a name and sales amount, but sets the commission rate to 0. The third constructor accepts a name and sets all the other fields to 0. Add the sales total from all the created objects and display the total to the screen.



```
file:///C:/Users/win10PC/documents/visual studio 2015/
Enter name of Salesperson name:
Jacobs
Enter Sales Amount:
250

Three Arguments constructor:
   The name of Sales Person: Jacobs
   Amount of sale = R250,00
   Commision =R50 000,00


Two arguments constructor:

   The name of Sales Person: Jacobs
   Amount of sale = R250,00
   Commision =R0,00


One Argument constructor
   The name of Sales Person: Jacobs
   Amount of sale = R0,00
   Commision   = R0,00
```

Answer:

using System;

```
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
namespace SalesTransactionDemo
{
public class Program
{
static void Main(string[] args)
{
SalesTransaxction st = new SalesTransaxction();
Console.WriteLine("Enter name of Salesperson name: ");
string name = Console.ReadLine();
Console.WriteLine("Enter Sales Amount: ");
double amount = Convert.ToDouble(Console.ReadLine());
Console.WriteLine("");
Console.WriteLine("Three Arguments constructor:");
Console.WriteLine(" Commision =" +
st.SalesTransaction(name, amount,
SalesTransaxction.commission_rate).ToString("c"));
Console.WriteLine("\n\n");
Console.WriteLine("Two arguments constructor:\n");
Console.WriteLine(" Commision =" +
st.SalesTransaction(name, amount).ToString("c"));
Console.WriteLine("\n\n");
Console.WriteLine("One Argument constructor ");
Console.WriteLine(" Commision = " +
st.SalesTransaction(name).ToString("c"));
Console.ReadKey();
}
}
public class SalesTransaxction {
string name;
double amount;
double commision;
public const double commission_rate=200;
public double SalesTransaction(string Name, double Amount,double
Commision_Rate)
{
Console.WriteLine(" The name of Sales Person: "+ Name);
Console.WriteLine(" Amount of sale = "+
Amount.ToString("c"));
commision = Amount * Commision_Rate;
return commision;
}
public double SalesTransaction(string Name, double Amount)
{
Console.WriteLine(" The name of Sales Person: " + Name);
Console.WriteLine(" Amount of sale = " +
Amount.ToString("c"));
commision = 0;
return commision;
}
public double SalesTransaction(string Name)
{
Console.WriteLine(" The name of Sales Person: " + Name);
Console.WriteLine(" Amount of sale = R0,00");
double commision=0;
return commision;
}
}
}
```

TOTAL MARKS: 100