

TA_Submission_Suren

by Suren Raj Tuladhar

Submission date: 10-Aug-2022 12:40PM (UTC+0100)

Submission ID: 184903887

File name: 54_Suren_Raj_Tuladhar_TA_Submission_Suren_5211_722883319.pdf (3.45M)

Word count: 7599

Character count: 40843

STUDENT NAME AND ID NUMBER	Suren Raj Tuladhar 202135813
Qualification	Pearson BTEC Level 5 Higher National Diploma in Digital Technologies (Cyber Security)
Academic Year	2021
Unit Number and Name	Unit 4 Programming
Unit Leader	Rajini Sachin
Unit Lecturer	Dr. Brinitha Raji
Assignment Title	Programming Paradigms
Type of Assignment	Business Report
Weighting	100%
Issue Date	01/07/2022
Formative Submission Date	21/07/2022
Summative Submission Date	11/08/2022
Assessor	Dr. Brinitha Raji
IV	Rajini Sachin
Resubmission Date	09/09/2022

Student Declaration**Student Declaration**

To protect the integrity and reputation of its qualifications, GBS requires that each Student taking PEARSON QUALIFICATION undertakes that this is the student's own work. A vital way of giving this assurance is for you to sign the authenticity statement below to confirm ownership of the work that may be presented to external examiner prior to certification. In so doing you are confirming that there has been no cheating or copying in producing the work and that where other sources of information have been used, they have been properly referenced.

Students should also be aware of the GBS guidance and regulations concerning plagiarism. All submissions should be your own, original work.

The Harvard Referencing System must be used. **The Wikipedia website must not be referenced in your work.**

You have the right to appeal against assessment decisions made regarding your submitted assignment.

Student Statement

To ensure certification of your achievement, before submitting your assignment for marking, please read and tick the box, then sign the box below.

My submitted assignment is my own work without any help except acceptable support from my Assessor(s)

³
By signing this document, I confirm I have read and understood the above Student declaration and verify my above Student statement regarding my submitted assignment to be true.

³

I confirm I have read and understood the above Student Declaration.

Student Name (print)	Suren Raj Tuladhar
Signature	
Date	11 th August 2022

Contents

Student Declaration	2
Unit 4	3
Programming	Error! Bookmark not defined.
5 LO1: Define basic algorithms to carry out an operation and outline the process.....	4
of programming an application	4
LO2: Explain the characteristics of procedural, object-orientated and event-driven Programming	11
LO3:	22
5 Implement basic algorithms in code using an IDE.....	22
LO4:	29
Determine the debugging process and explain the importance of a coding standard	30

3

LO1: Define basic algorithms to carry out an operation and outline the process of programming an application

Vocational Scenario and Role Play:

As a key player in the education market, I must give proper knowledge to the students regarding information, technology/computer science. Our team's main agenda is to provide teaching material to the student of Level 4 HNC & Level 5 HND. I have prepared all of the projects according to it and I believe the student will get a good understanding of the programming.

Programming:

Programming is the create a set of instructions which guides a computer on how to perform a task. This set of instructions is usually written in the programming language. Hence, I have listed the are 4 types of programming languages which are as follows.

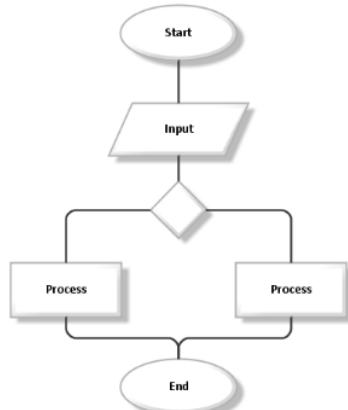
15

- Procedural Programming Language
- Functional Programming Language
- Scripting Programming Language
- Logic Programming Language
- Object-Oriented Programming Language

Definition of Algorithms: Algorithms can be defined as the set of sequential instructions in a flow to solve or complete a specific task. An algorithm can be designed by using pseudocode or flowcharts. Examples: the procedure for baking a cake, the process of doing laundry, driving a car etc. Hence, the set of algorithms can be further broken down into flowcharts and pseudocode.

9

Definition of Flowcharts: A flowchart is a formalized graphic representation of a logic sequence, work or manufacturing process, organization chart, or similar formalized structure. The purpose of a flow chart is to provide people with a common language or reference point when dealing with a project or process.



Here, the above diagram is for the flowchart, where a set of instructions are represented in graphical form. Where a set of instructions are mentioned in their respective geometrical shapes.

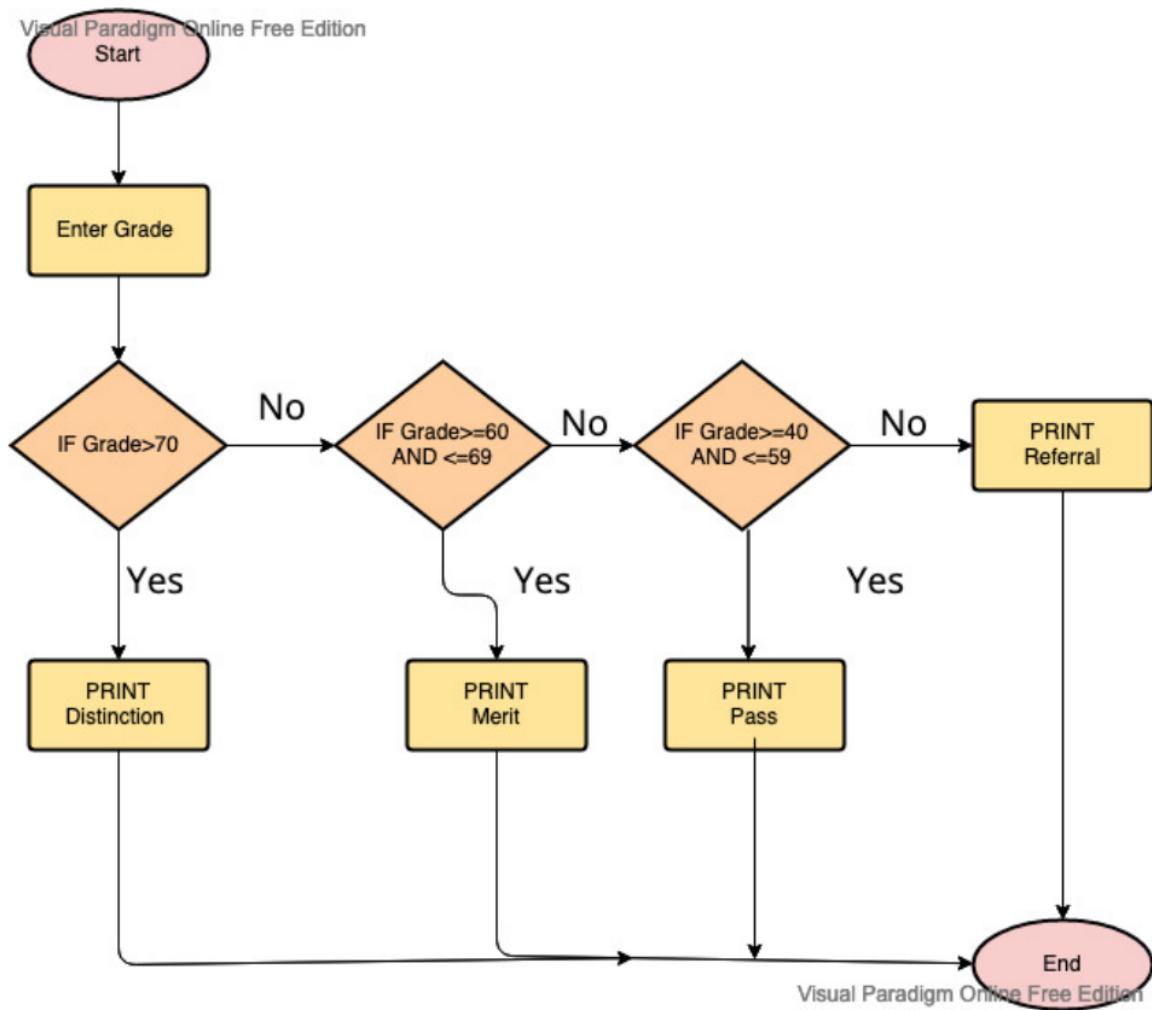
Definition of Pseudocode: It can be defined as a step of instructions written in a set of sequences which helps programmers develop algorithms. The pattern of pseudocode is simple. For example,

```
IF spaceship sprite touches asteroid sprite THEN
    show explosion sprite
    play explosion sound
    subtract a life
END IF

IF lives = 0 THEN
    stop game
    show game over screen
ELSE
    restart game
ENDIF
```

Further examples are mentioned below:

CALCULATE STUDENT GRADE

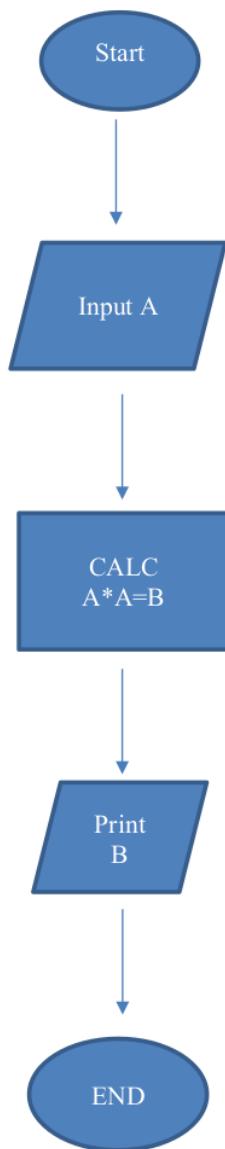


```

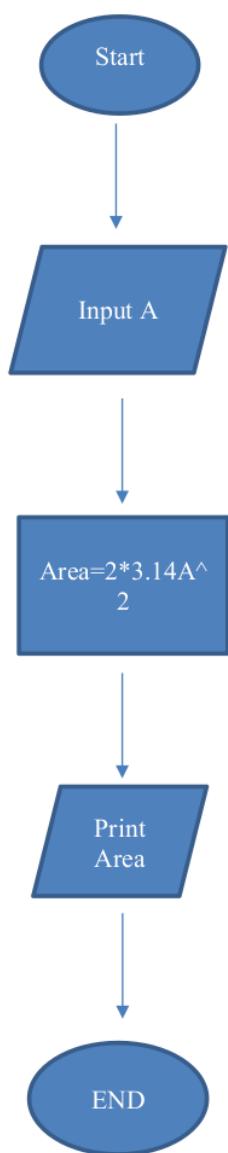
BEGIN
DECLARE Grade
INPUT Grade
IF (Grade>70) THEN OUTPUT Distinction
ELSEIF (Grade>=60 AND <=69) THEN
OUTPUT Merit
ELSEIF (Grade>=40 AND <= 59) THEN
OUTPUT Pass
ELSE
OUTPUT Referral
ENDIF
END
  
```

Calculate The Square

```
BEGIN  
DECLARE A, square  
INPUT A  
STORE A  
square= A*A  
STORE square  
OUTPUT square  
End
```



```
BEGIN  
DECLARE A, area  
INPUT A  
STORE A  
Area = 2*3.14*A^2  
PRINT Area  
END
```

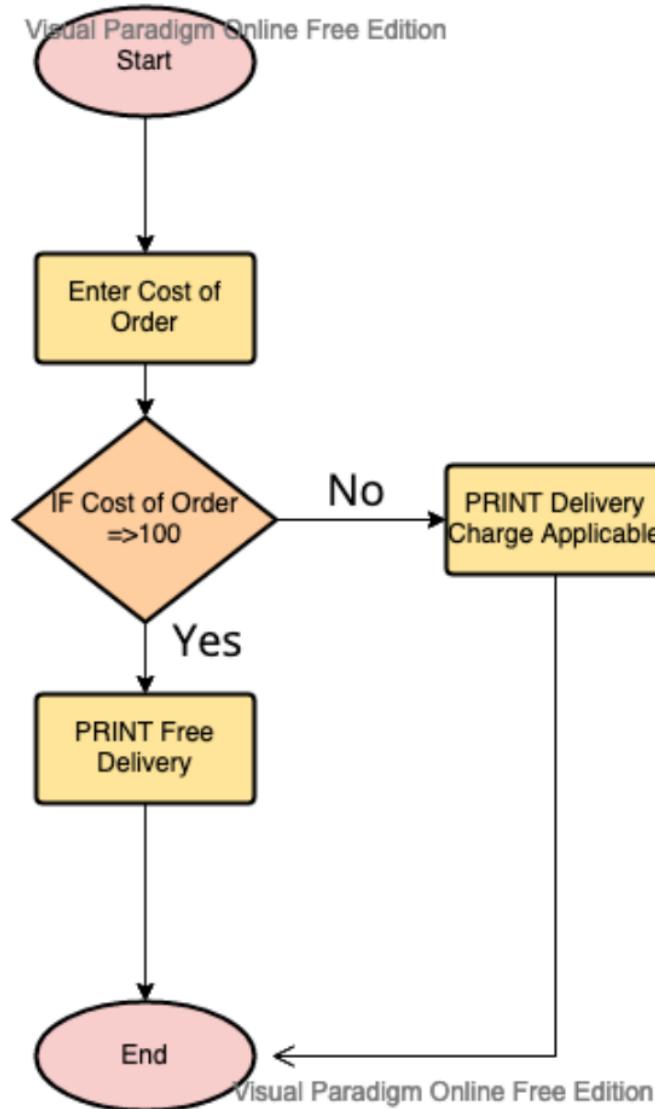


Calculate Free Delivery Orders

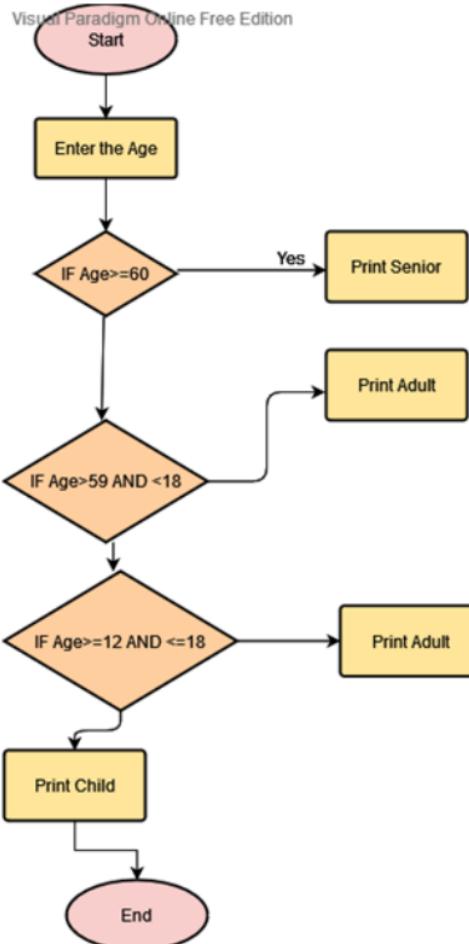
```

BEGIN
DECLARE O,C ost of Order
Input Cost Of Order
IF Cost =\$100
PRINT Free Delivery
ELSE,
PRINT Delivery Charges Applicable
End If
END

```



Calculate the Age & Ticket

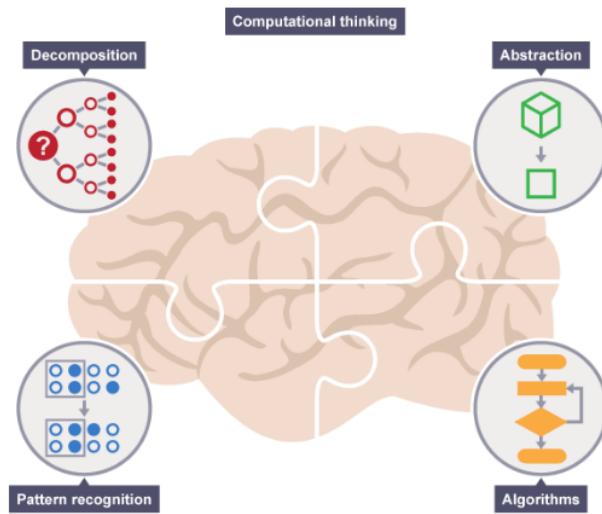


```

BEGIN
DECLARE Age
INPUT Age
IF (Age>60) THEN
OUTPUT SENIOR
ELSEIF (Age >=59 AND <18) THEN
OUTPUT Adult
ELSEIF (Age >= 12 AND <=18) THEN
OUTPUT STUDENT
ELSE
OUTPUT CHILD
ENDIF
END
  
```

M1. The Process of Building An Application/Program

The steps are very crucial to executing the programs. No wonder in the flowchart and pseudocode these sequential steps must take to execute the operation successfully.



Computational Thinking: A problem can be solved with the help of a computer. However, the process of execution should be known before a problem can be tackled. the problem itself and how it could be solved need to be understood. Computational thinking also allows us to take on a complex problem, understand what the problem is and develop possible solutions. Afterwards, It can present these solutions in such a way that both i.e. a computer and a human, can understand. The following are the four cornerstones of computational thinking.

- Decomposition
- Recognition patterns
- Abstraction
- Algorithms
- Code

1. **Decomposition:** The 1st phase of any program is to simplify the problem. In other words, break down a complex problem or a system into smaller which is more manageable parts or structures to process.
2. **Recognition Patterns:** Pattern recognition is a very crucial part of application building. Hence it is responsible for identifying the similarities among and within the problems. Rather than repeating the program for similar programs, it's better to have a common solution for most problems with similar patterns.
3. **Abstraction:** Time is a crucial element for everyone. Even for the programmer, it is very crucial to focus on the important information only. Irrelevant details are time lechers.

- 4. Algorithms:** Programming is all about the step-by-step solution to a problem. In simpler words, it can be referred to as the rules to follow to solve problems. A flowchart or pseudo-code can be a good example of the representation of the algorithms.
- 5. Code:** Coding is fundamental the program development. Good and organized coding is appreciated to develop any sort of program and way for programming.

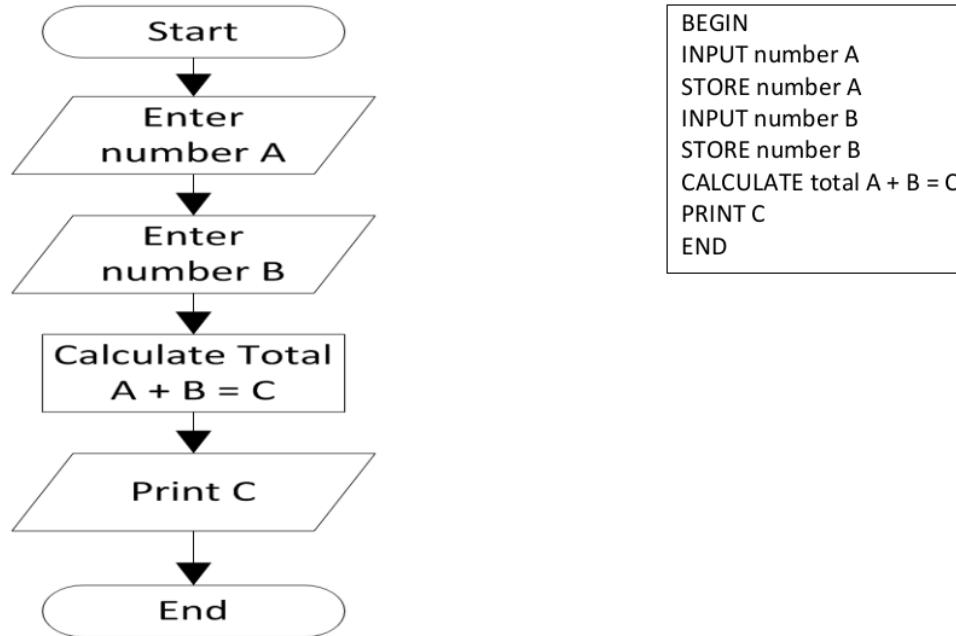
Conclusion: In conclusion, each cornerstone is as important. They are the legs of the table, If any one leg is missing or it's not strong the entire table won't be able to stay up and eventually it will collapse. So it is always advised able to any programmer to follow the basics of programming which will lead to a better program.

- **Step 1:** The first step is to understand the basic operation. Without these rudimentary essentials, the ignition of the pseudocode won't be is possible.

Pseudocode Operations

Operation		
BEGIN	BEGIN denotes the start of the program END denotes when the program will end	BEGIN INPUT length of room INPUT width of room CALCULATE area = length * width END
END		
INPUT	INPUT is used to denote when a user has entered something into the program OUTPUT is used to output/display information to screen	BEGIN INPUT number CALCULATE square = number * number OUTPUT square END
OUTPUT/PRINT		
<i>READ and WRITE are used in file handling</i>		
READ	READ is used where you wish to inspect the contents of a file or folder ie read a record into the variable record.	BEGIN READ MyFile.txt Record END
WRITE	WRITE is used when you wish to add to a file. The WRITE statement writes the record to the file.	BEGIN WRITE MyFile.txt Answer1, Answer2, 'abc123' END

- **Step 2:** Every operation should be in the sequence form to complete the operation or the task. For example, the pseudocode should be written sequentially while the flowchart should follow the same patterns.



- **Step 3:** After the sequence, we have to be concerned about the structure of the code and flowchart.

```

CALCULATE GRADE
BEGIN
  IF grade is greater than or equal to 75 THEN
    OUTPUT "Well done"
  ENDIF
  INPUT grade
END
  
```

The hierarchy of the program is incorrect because INPUT grade should appear before the if statement

the above code, we can see the error that the “INPUT grade”

Here, In

- **Step 4:** Followed by structure, Indentation comes into play. Where by indenting the code will be much more readable and easy to follow. Also, one can identify if the instruction is in the right order or not. For example.

```
CALCULATE STUDENT GRADE
BEGIN
    IF grade greater than or equal to 70 THEN
        OUTPUT Distinction
    ELSEIF grade between 60 and 69 THEN
        OUTPUT Merit
    ELSEIF grade between 40 and 59 THEN
        OUTPUT Pass
    ELSE
        OUTPUT Referral
    ENDIF
END
```

- **Compilation and compilers:**
- **Linkers:**
- **And how the source code is converted into the executable program:**

3

LO2: Explain the characteristics of procedural, object-orientated and event-driven Programming

Programming Paradigms

It is the way of programming or can be classified as an approach to solving problems by using programming languages. It does not refer to any specific programming language however it does refer to the way programs. In other words, programming paradigms eliminate the complexity of programs. The various programming paradigms are:

- Procedural Programming Paradigms
- Object-Oriented Programming Paradigms
- Event-Driven Programming Paradigms



Procedural Programming Paradigm:

It is the programming language that is based on structured programming. The code in POP is written to break down the tasks into data structures, subroutines, and a collection of variables. The POP paradigm is based on the concept of the procedure call. Procedures are simply a series of steps to be followed.

Procedural Programming:

In simple terms, this programming is simply a series of steps which is broken down into different tasks which need to be followed. POP paradigm is based on the procedure call concept. This paradigm treats data and procedures as two different entities. A linear top-down approach can be observed in this paradigm to solve the problem. Which is later categorised as a series of steps which need to be carried out to solve the problem. For example, C, Basics, Pascal etc.

8

Characteristics of POP

- **Predefined Functions:** In higher-level programming languages a predefined function is generally an instruction which is built in the form of a registry or Library. For example “CharAT()” is which

searches for a character position in a string

- **Local Variable:** Generally, as the name suggests the local mean unbounded and variable denotes any variable which is declared in the framework. As result, the local can be used in the method where it is initially pre-called. An example of a local variable in the code below is age,

```
public class Main {  
    public void Age() {  
        int age = 0;  
        age = age + 18;  
        System.out.println("Your age is: " +age);  
    }  
    public static void main (String args[]) {  
        main m = new Main();  
        m.Age();  
    }  
}
```

(Key Features of Event-Driven Programs., 2022)

- **Global Variable:** Unlike a local variable global variable can be used in all functions. Generally, it is called outside of every function which is defined in the code. An example of a global variable is:

```
int year;  
int main(){  
    year = 2014;  
    int newyear = year + 1;  
} (Key Features of Event-Driven Programs., 2022)
```

- **Parameter Passing:** Parameter passing is generally used by the function as it allows a value to be passed through a program. For example

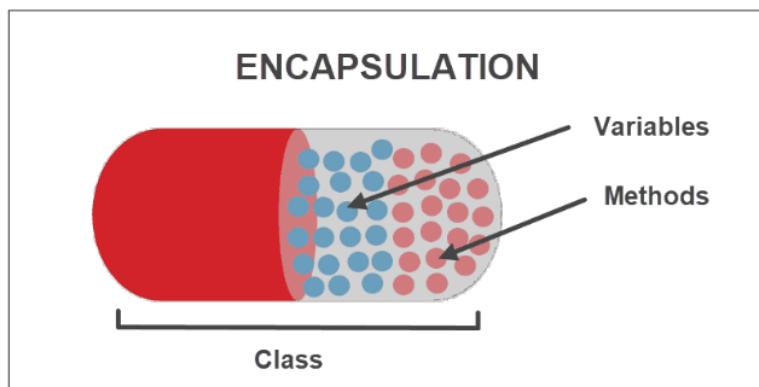
```
Area ( int l, Int w)  
String Name  
Int Length, Width,Radius,area  
{ area= Length X width  
print (area)  
}  
Area (Int R)  
{area = 2πr^2  
Print (area)  
}
```

Object-Oriented Programming Paradigm

Also known as OOP, it is another type of high-level programming language based on the concept. Being one of the most widely used programming languages in the world it is often known as the blueprint of “object”, which often contains data or attributes. Example Java, C++, C#, PHP<JavaScript etc.

Characteristics of OOPP

- **Encapsulation Enforces Modularity:** In short, encapsulation is capturing data safe keeping it from the outside interface.

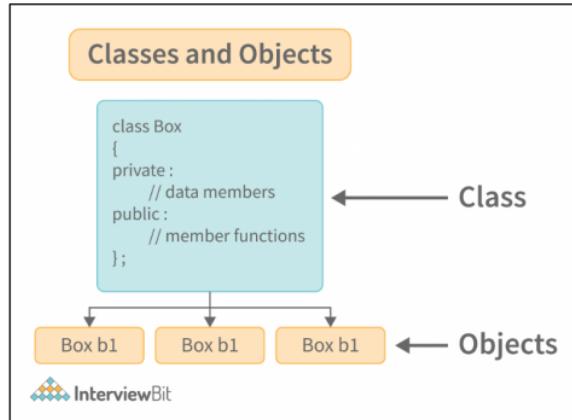


⁶ As an example, a car has descriptive characteristics and actions. Even though the colour of the car can be changed the model or make are examples of properties that cannot change. Here, a class **encapsulates** all the information into the singularity, in which some elements are modifiable while some may not. (home, support and </form>, 2022).

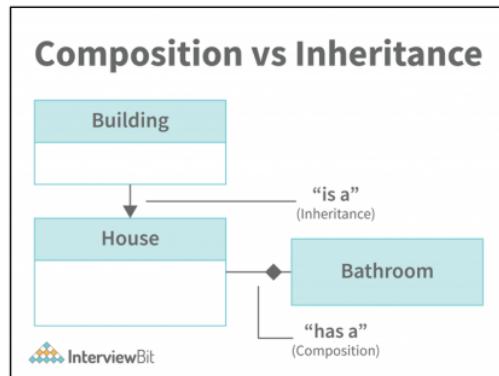
- **Abstraction:** In simple words, it is the act of representing the key features without including the full background details or any explanations. This process ensures that it creates reusable objects which can be further designed to be common in business logic and standardize both at the same time. As result, they can be used consistently in many different applications and reduces complexity.

⁶ For example, A computer to connect to a different network. A web browser needs an internet connection. However, the connection type seems to be irrelevant as an established connection to the internet represents an **abstraction**. Hence the abstraction can be managed through the use of hierarchical classification.

- **Class:** The class can be defined as the blueprint for an object. Hence, the blueprint can be defined as the methods, attributes and other features and attributes of the objects. In other words, the origin of the object can be derived from the classes. (Top Characteristics of Object-Oriented Programming, 2022)



- **Inheritance & Composition:** Inheritance can be defined as the process of replication of an object to create or inherit its features, and methods. Whereas composition is the process of combining multiple objects to create a new and unique object. Combining objects and inheriting their aspects, Properties is one of the most important aspects of OOP. (Top Characteristics of Object-Oriented Programming, 2022)



- **Polymorphism:** This word is derived from Greek, which states that “many forms”. A unique feature that allows one interface to be used for a general class of actions i.e. “one interface, multiple methods”. (CSC238 INTRODUCTION TO OOP Chapter ppt video online download, 2022)

Event-Driven Programming Languages:

The event-driven programming paradigm uses the occurrence of events to determine the control flow of the program. Such a type of application uses the callback function. Generally, the event-driven style of programming is supported by all programming languages in which there is a main loop feature in the much event-driven application which is used to listen for events and also listen to the call back function when such event is detected. Such programming is used in the following cases:

- When programming games in which a set of objects carries out control flow.
- When creating server applications, it is undesirable to spawn service processes if, for one reason or another.
- When building graphical user interfaces. (Event-Driven Programming - Applications & Features, 2022)

12

Characteristics of Event-Driven Programming:

- **Service Oriented:** One of the key features of this event-driven programming paradigm is service oriented. If you have to write programs made for services then you have to use this programming paradigm. These programs only take very small system resources as result it's run in the background of the operating system.
- **Time Driven:** One of the key features of this event-driven programming paradigm is service-oriented is Time driven in event-driven programming. The code can be programmed to run on a time trigger. it's a pre-set to do a task. For example, we can take the windows update as the example of time driven, in which the downloadable update is usually set by the user.
- **Events:** Events include mouse, keyboard and user interface, which events need to be triggered in the program to happen, with the interaction of the user with an object, for example, clicking a button by a mouse, using the keyboard to select a button, etc.
- **Event Handlers:** Event handlers are a type of function which run a specific action for a specific event. For example, it could be a button that when the user clicks a pop-up message will be displayed for the user and it will close the message when the user clicks on the close button. (Event-Driven Programming Features, 2022)
- **Trigger Functions:** In event-driven programming, trigger functions are functions that decide what code to execute when there is a specific event occurs, which are also used to select which event handler to use for the event when there is a specific event occurred.

Conclusion: In conclusion, these 3 programming paradigms are equally important in their respective sectors. Procedural Programming follows top-down approach whiles, whereas OOPP follows a bottom-up approach. OOPP Is much more secure than POPP, Whereas in POPP the execution of the program statement is predetermined however in OOPP it is designed using the concept of an object which interacts with the real world. Talking about the EDPP it is determined by events such as user actions on the system. Where program execution is mostly determined by the system. In which one can define what will occur when a user executes an event.

M2

Comparison of POP, OOPP and Event-driven Programming Paradigms

Comparing & Contrasting Different Programming Paradigms

Procedural Programming Paradigm: Procedural programming is based on structured programming in the source code procedure-oriented programming is written so that it helps to simplify, and divide the tasks into subroutines, variables & data structures. POP works on the direction so that the program can follow procedures to accomplish the tasks by providing instructions to the system.

- Program is divided into small parts which we know as functions.
- POP follows a top-down approach.
- No access specifier in procedural programming.
- New data and functions are hard to add.
- POP is less secure. 8
- Overloading the program is not possible.
- In POP function is more important than data and it is based on the unreal world.
Examples: C, FORTRAN, Pascal, Basic etc.
(Procedural, Object-oriented and event-driven Programming Languages (Visual Basic). - Boot Poot, 2022)

As large & complex software development comes into play it generally requires OOP & EDP. The object-oriented programming paradigm is a high-level programming language & is based on object concepts.

Object Oriented Programming: OOP works into instances of classes which is known as the blueprint of objects. The object in OOP contains data in field forms called attributes or properties & code is written in procedure form called methods or functions.

- The OOP program is divided into parts known as objects.
- It follows a bottom-up approach.
- It has access specifiers like private, public, protected etc.
- It can add new data and functions with ease.
- It has much more security than PP.
- It is prone to overloading.
- Based on the real world.
- Example: C++, Java, Python, C# etc.

Event-driven programming paradigms: It uses the events program's control flow. Such applications are programmed to detect events and when such events occur it uses event handling procedures appropriately similar to the call back function or method. The events determine the flow of the program such as user actions, user interactions, and sensor inputs & such program execution is determined by the system as what will occur is defined based on the user executing the certain event. EDP programming style is supported by all programming languages but the way of implementation is different.

- General purpose programming
- Code is simple & easier to implement for the compliers & interpreters while tracking the flow of the program.
- EDP has more benefits & more advantages than overusing POP.
- EDP allows the reusability of code.
- Can be executed through a single event. For example, instances like mouse clicks or press of keys (Object-Oriented, Event-Driven and Procedural Programming - 1263 Words | Report Example, 202

Event Driven Programming Paradigm	Procedural Programming Paradigm	Object Oriented Programming Paradigm
It provides GUI interface to code programs	It has CUI to code the programs	It uses modules to write code.
The actions are denoted on events. Such even can be trigger by clicking mouse or from the keyboard strokes	Commands are executed in liner fashion an executed in similar way.	Objects and functions are polished for the interaction to perform a certain task.
Prioritized on selecting user interface	Prioritized sequential execution of steps.	Prioritized on objects or data and facilities and secure them with good securities
Uses Visual basic and C#	Uses Basic, Fortran and COBOL	Uses Smalltalk,C++ anda Java

Conclusion

Programming paradigms are the essential sector where change is constantly required in daily computer usage. Where, OOP & EDP enables users to provide multiple, complex inputs even externally as needed... POP has inferior user comp ability compared to OOP & EDP as they provide features like reusability of codes for similar functions. Whether it is procedure-oriented programming, object-oriented programming, or event-driven programming, every programming paradigm has its pros and cons built in them however each of them helps to make programs simpler and user-friendly reducing the complexity of programs which is the sole purpose of all these different programming paradigms.

3
LO3:**Implement basic algorithms in code using an IDE**

To show the implementation of a program Using An IDE. I have made 3 programs which will be demonstrated below to show.

- I. Switchback Nepal MTB Service Information System.
- II. VOX Cinema Ticket System
- III. Zomato Delivery System

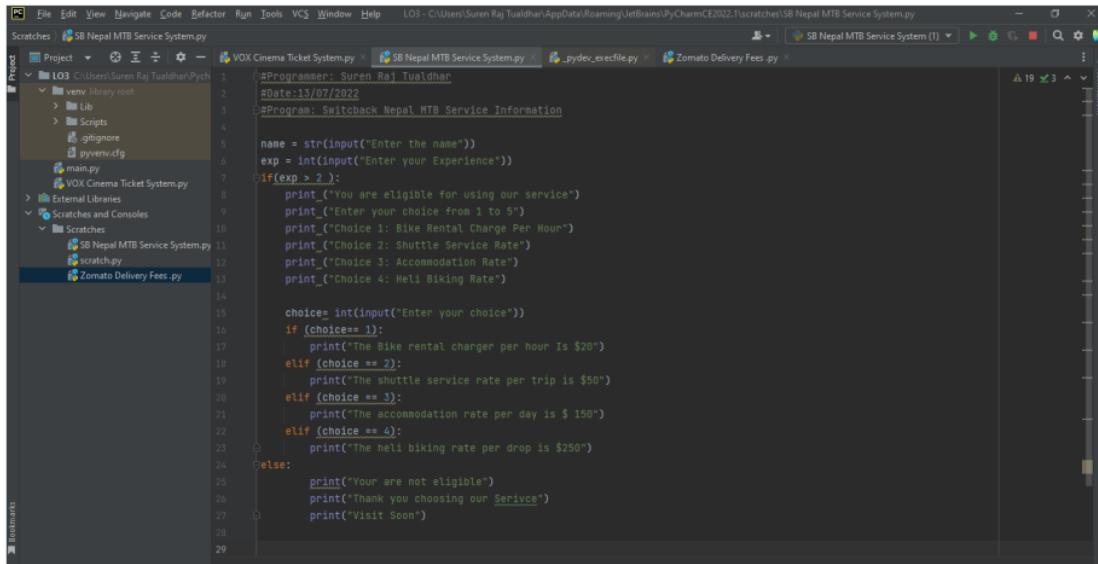
- **Switchback Nepal MTB Service Information System:** Mountain biking is an extreme sport which been gaining popularity in a short period. This sport has created the opportunity for local and international markets as well. As per the geography of Nepal, many riders across the world come to ride in the Himalayas. To relay information in a short span of per with iod great efficiency I have to develop this program which gives service information according to the user inputs.
As for the initial steps we will be starting by making the pseudocode for the program. I have come up with simple codes as far as possible.

```
BEGIN
DECLARE Name
DECLARE Question
DECLARE Choice
INPUT Name
INPUT Experience? Yes/No
IF (Experience >= 5) Then
IF (Question == Yes) THEN
OUTPUT      ("You are eligible for using our service")
            ("Enter your choice from 1 to 5")
            ("Choice 1: Bike Rental Charge Per Hour ")
            ("Choice 2: Shuttle Service Rate")
            ("Choice 3: Accommodation Rate ")
            ("Choice 4: Heli Biking Rate")

INPUT Choice

IF (Choice == 1) THEN
    OUTPUT ("The Bike rental charger per hour Is $20")
ELSE IF (Choice == 2) THEN
    OUTPUT ("The shuttle service rate per trip is $50")
ELSE IF (Choice == 3) THEN
    OUTPUT ("The accommodation rate per day is $ 150")
ELSE IF (Choice == 4) THEN
    OUTPUT ("The heli biking rate per drop is $250")
ENDIF
END
```

As per this pseudo code, the algorithm is designed in such a way that first the input is taken from the Users. I.e., Name and their riding experience. Then relative to the input the output is presented to the user in the form of a series of choices.



```

Project : SB Nepal MTB Service System
Scratches : VOX Cinema Ticket System.py, SB Nepal MTB Service System.py, _pydev_execfile.py, Zomato Delivery Fees.py
LO3 : venv, library root, Lib, Scripts, .gitignore, pyenv.cfg, main.py, VOX Cinema Ticket System.py, External Libraries, Scratches, SB Nepal MTB Service System.py, scratch.py, Zomato Delivery Fees.py

#Programmer: Suren Raj Tualdhar
#Date:13/07/2022
#Program: Switchback Nepal MTB Service Information

name = str(input("Enter the name"))
exp = int(input("Enter your Experience"))

if(exp > 2):
    print("You are eligible for using our service")
    print_("Enter your choice from 1 to 5")
    print_("Choice 1: Bike Rental Charge Per Hour")
    print_("Choice 2: Shuttle Service Rate")
    print_("Choice 3: Accommodation Rate")
    print_("Choice 4: Heli Biking Rate")

    choice_ = int(input("Enter your choice"))
    if (choice_== 1):
        print("The Bike rental charger per hour Is $20")
    elif (choice_ == 2):
        print("The shuttle service rate per trip is $50")
    elif (choice_ == 3):
        print("The accommodation rate per day is $ 150")
    elif (choice_ == 4):
        print("The heli biking rate per drop is $250")
    else:
        print("Your are not eligible")
        print("Thank you choosing our Service")
        print("Visit Soon")

```

Fig, Source Code.

Here, In this figure, you can see the source of the program.



```

Run : SB Nepal MTB Service System (1)
C:\Users\Suren Raj Tualdhar\PycharmProjects\LO3\venv\Scripts\python.exe "C:/Users/Suren Raj Tualdhar/AppData/Roaming/JetBrains/PyCharmCE2022.1/scratches/SB Nepal MTB Service System.py"
Enter the name:Suresh
Enter your Experience:2
You are eligible for using our service
Enter your choice from 1 to 5
Choice 1: Bike Rental Charge Per Hour
Choice 2: Shuttle Service Rate
Choice 3: Accommodation Rate
Choice 4: Heli Biking Rate
Enter your choice
The Bike rental charger per hour Is $20
Process finished with exit code 0

```

Fig Output

Here, in the above figure, we can see the execution of the program without any error. Where output shows that the customer belongs to the senior category.

The screenshot shows a PyCharm IDE interface with multiple files open in the background. In the foreground, the 'SB Nepal MTB Service System.py' file is being edited. A red box highlights line 25 of the code, which contains a syntax error. The error message 'SyntaxError: invalid syntax' is displayed in the 'Run' tab at the bottom.

```

print("Choice 1: Bike rental")
print("Choice 2: Shuttle Service Rate")
print("Choice 3: Accommodation Rate")
print("Choice 4: Heli Biking Rate")

choice= int(input("Enter your choice"))
if (choice==1):
    print("The Bike rental charger per hour Is $20")
elif (choice == 2):
    print("The shuttle service rate per trip is $50")
elif (choice == 3):
    print("The accommodation rate per day is $ 150")
elif (choice == 4):
    print("The heli biking rate per drop is $250")
else:
    print("Your are not eligible")
    print("Thank you choosing our Service")
    print("Visit Soon")

```

Fig, Error and Bug Fix.

In the above diagram, we can see that there is a “Syntax Error” displayed in the output. In the source code, I have intentionally removed (") in Line 25 which I have marked. This shows the importance of the small details in the programming. Where programmers should mindful when making the program.

- **VOX Cinema Ticket System:** VOX is a popular movie theatre here in Dubai. According to their ticketing system, there are certain categories based on which they separate their audience by. Based on the age they have their audience categories.

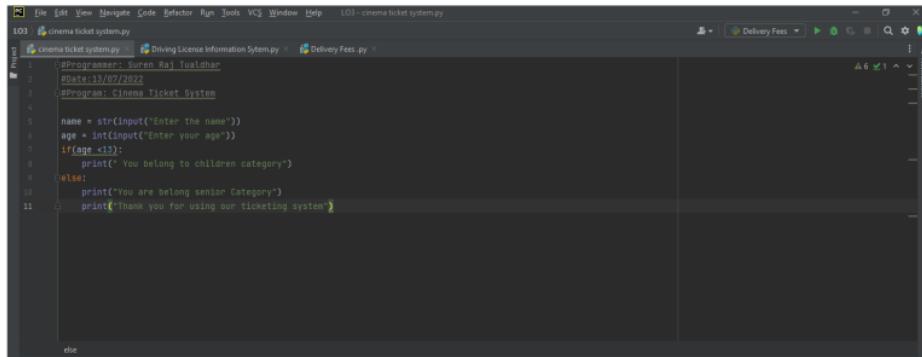
```

BEGIN
DECLARE Age
INPUT Age
IF (Age>60) THEN
OUTPUT SENIOR
ELSEIF (Age >=59 AND <18) THEN
OUTPUT Adult
ELSEIF (Grade >= 12 AND <=18) THEN
OUTPUT STUDENT
ELSE
OUTPUT CHILD
ENDIF
END

```

As mentioned above, The ticket system works based on age number. Here, as a prototype, if the age is greater than or equal to 60 then it would categorize the audience as a Senior. Likewise, if the age is greater than between 18-59 then it would categorize the audience as Adults.

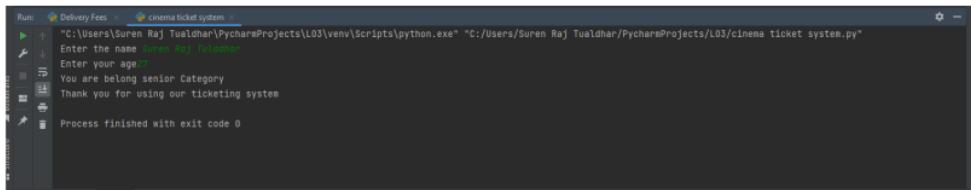
If the age is between 12-18 then it would categorize the audience as Students or above else it would display Child if the other conditions are not met.



```
1 #!/usr/bin/python
2 #programmed: Suren Raj Tualdhar
3 #Date: 15/07/2022
4 #Program: Cinema Ticket System
5
6 name = str(input("Enter the name"))
7 age = int(input("Enter your age"))
8
9 if(age <12):
10     print(" You belong to children category")
11 else:
12     print("You are belong senior Category")
13
14 print("Thank you for using our ticketing system")
```

Fig, Source Code.

Here, In this figure, you can see the source of the program.



```
Run: Delivery Fees x cinema ticket system x
1 "C:/Users/Suren Raj Tualdhar/PycharmProjects/L03/venv/Scripts/python.exe" "C:/Users/Suren Raj Tualdhar/PycharmProjects/L03/cinema ticket system.py"
2 Enter the name Suren Raj Tualdhar
3 Enter your age 25
4 You are belong senior Category
5 Thank you for using our ticketing system
6
7 Process finished with exit code 0
```

Fig Output

Here, in the above figure, we can see the execution of the program without any error. Where output shows that the customer belongs to the senior category.

The screenshot shows a PyCharm IDE interface. The code editor displays a Python script named 'cinema ticket system.py'. The code contains a conditional block with a syntax error:`1 #Programmer: Suren Raj Tauldar
2 #Date:13/07/2022
3 #Program: Cinema Ticket System
4
5 name = str(input("Enter the name"))
6 age = int(input("Enter your age"))
7 if(age >3):
8 print(" You belong to children category")
9 else:
10 print("You are sbelong senior Category")
11
12 print("Thank you for using our ticketing system")`The line 'if(age >3):' has a yellow box around it, and the character '>' is highlighted with a red circle. The run output shows the error:`Run: Delivery Fees cinema ticket system
File "C:/Users/Suren Raj Tauldar/PycharmProjects/L03/venv/Scripts/python.exe" "C:/Users/suren Raj Tauldar/PycharmProjects/L03/cinema ticket system.py"
line 7
 if(age >3):
 ^
SyntaxError: invalid syntax
Process finished with exit code 1`

Fig, Error and Bug Fix.

In the above diagram, we can see that there is a “Syntax Error” displayed in the output. In the source code, I have intentionally removed “>” in Line 7 which I have marked. This shows the importance of the small details in the programming. Where programmers should be mindful when making the program.

- **Zomato Food Delivery System:** Zomato is the most popular food delivery system in UAE. On the occasion of Eid a holy festival in the country. Zomato has announced a free delivery service across the 7 states in UAE. Based on that I have designed the prototype program which will distinguish the potential customer who is eligible for the service.

```
BEGIN
DECLARE Name
DECLARE Address
DECLARE Amount
INPUT Name
INPUT Address
INPUT Amount
IF (Amount >100) THEN
    OUTPUT ("Congratulation")
    ("Your food will be delivered in 30 mins")
    ("Enjoy our free delivery service")
    ("Bon Appetite")
    ("For More Contact Us @ +971 524707211")

ELSE
    OUTPUT ("Congratulation")
    ("Your food will be delivered in 30 mins")
    ("Please ensure minimum Purchase order is 100AED")
    ("To Enjoy our free delivery service")
    ("Bon Appetite")
    ("For More Contact Us @ +971 524707211")
ENDIF
END
```

As mentioned above, the food delivery system works based on the minimum purchase order. Here, as a prototype, if the purchase order is ≥ 100 then it would categorize the customer as eligible for the free delivery service. Else it encourages the customer to maintain a minimum purchase order to enjoy free delivery service.

```

File Edit View Navigate Code Behavior Run Tools VCS Window Help L03 - C:/Users/Suren Raj Tualdhar/AppData/Roaming/JetBrains/PyCharmCE2022.1/scratches/Zomato Delivery Fees.py
Scratches Zomato Delivery Fees.py RTA Driving License Information System.py Zomato Delivery Fees.py
Project VOX Cinema Ticket System.py
1 #Programmer: Suren Raj Tualdhar
2 #Date:18/07/2022
3 #Program: Food Delivery Fees
4
5 name = str(input("Enter the name"))
6 address = str(input("Enter your Delivery Address"))
7 amount = int(input("Enter Total Payable Amount"))
8 if(amount >100):
9     print("Congratulation ")
10    print("You are food will be delivered in 30min")
11    print("Enjoy our free Delivery Service")
12    print("Bon Appetite ")
13 else:
14     print("Congratulation")
15    print("You are food will be delivered in 30min")
16    print("To Enjoy our Free Delivery Service")
17    print("Minimum Purchase Order AED 100")
18    print("Bon Appetite ")
19
20 print("For More Contact Us @ +971 524707211 ")
21

```

Fig, Source Code.

Here, In this figure, you can see the source of the program.

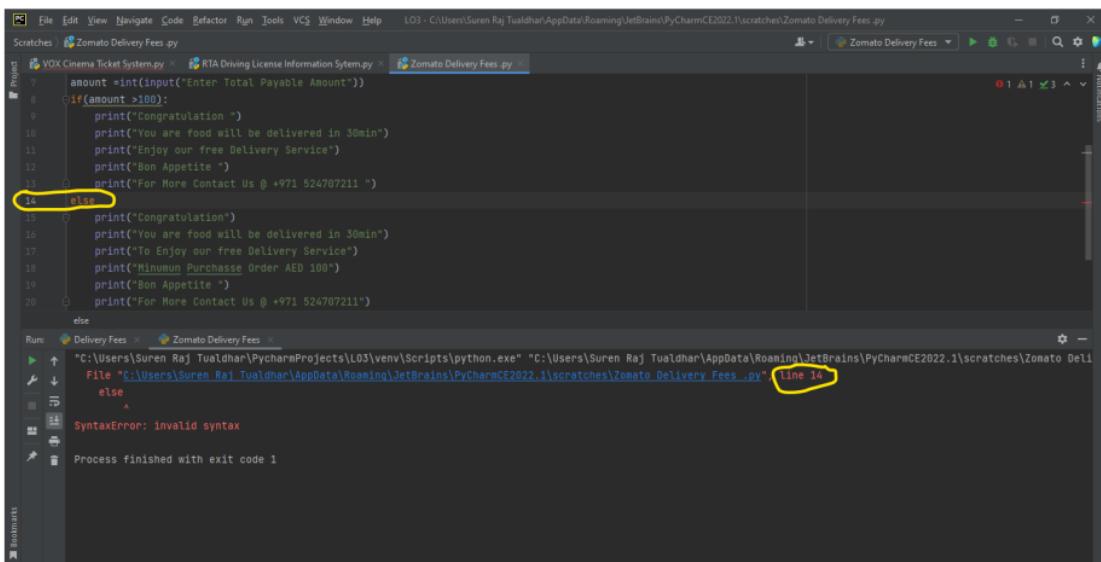
```

Run: Delivery Fees x Driving License Information System x
C:\Users\Suren Raj Tualdhar\PycharmProjects\L03\venv\Scripts\python.exe "C:\Users\Suren Raj Tualdhar\AppData\Roaming\JetBrains\PyCharmCE2022.1\scratches\Driving Lic
Enter the name Suren Raj Tualdhar
Enter your age?
YOU ARE ELIGIBLE FOR USING dl SYSTEM
Enter your choice from 1 to 5
Choice 1: Opening A File
Choice 2: Issuing License
Choice 3: Handbook Manual
Choice 4: Knowledge and Innovation
Enter your choice:
Your Opening File fee is 200AED
Process finished with exit code 0

```

Fig Output

Here, in the above figure, we can see the execution of the program without any error. Where output shows



The screenshot shows the PyCharm IDE interface. In the top navigation bar, the file 'Zomato Delivery Fees.py' is selected. The code editor displays the following Python script:

```

amount =int(input("Enter Total Payable Amount"))
if(amount >100):
    print("Congratulation ")
    print("you are food will be delivered in 30min")
    print("Enjoy our free Delivery Service")
    print("Bon Appetite ")
    print("For More Contact Us @ +971 524707211 ")
else:
    print("Congratulation")
    print("You are food will be delivered in 30min")
    print("To Enjoy our free Delivery Service")
    print("Minimum Purchasse Order AED 100")
    print("Bon Appetite ")
    print("For More Contact Us @ +971 524707211")

```

A yellow box highlights the line 'else:' at the beginning of the second conditional block. Another yellow box highlights the word 'Line 14' in the error message 'File "C:/Users/Suren Raj Tualdhar/AppData/Roaming/JetBrains/PyCharmCE2022.1/scratches/Zomato_Delivery_Fees.py" Line 14'. The run output window shows the error: 'SyntaxError: invalid syntax'.

Fig, Error and Bug Fix.

In the above diagram, we can see that there is a “Syntax Error” displayed in the output. In the source code, I have intentionally removed “:” in Line 14 which I have marked. This shows the importance of the small details in the programming. Where programmers should mindfull when coding the program.

5
LO4:**Determine the debugging process and explain the importance of a coding standard**

Debugging Definition: The process of fixing a bug in the software is known as debugging. It generally includes identifying, analyzing and eliminating errors and bugs. It generally comes into play when the software fails to execute properly which results in the crashing of the software. To prevent debugging, come to the rescue by removing all the potential errors. The stages of debugging i.e. identification of the errors which have to be corrected make the process complex and time-consuming.

Importance of Debugging:

Debugging comes into action from the very beginning of the software development lifecycle. It is mostly used due to its following.

- **Time-Saving:** Debugging saves time as the errors and bugs are immediately reported
- **Stress-free:** Developers don't need to search for the bugs manually as well as the search is eliminated
- **Structured:** The debugging is reported systematically, making it easier to find and solve the problems.
- **Smooth-running:** The entire process is made smooth as the errors and bugs are identified and fixed instantly. As a result, as it makes things easier for the developer to work on the project much more smoothly.
- **Cost-effective:** Debugging helps reduce the cost of software developments by preventing software crashes in real-time.
- **Productive:** To increase the efficiency of the whole development software development team and its life cycle debugging comes to the

Error & Types in Debugging:

When it comes to debugging there are mostly 3 types of error a developer faces in the process which are

- Syntax Error
- Run-Time Error
- Semantic Errors

- Syntax Error:** To run a computer program, it should be free from errors and bugs from the deep root i.e. from the source code. A syntax error is when mistakes are made while writing the program code. Such error or code made it difficult to process, leading to obstruction in computer program execution. As translators cannot execute the program and the program is terminated mid-way due to program code errors like spelling mistakes, improper use of cases, improper punctuation, missing brackets, etc. (Mishra, 2022)

```

File Edit View Navigate Code Refactor Run Tools VCS Window Help LO3 - C:\Users\Suren Raj Tualdhari\AppData\Roaming\JetBrains\PyCharmCE2022.1\scratches\Zomato Delivery Fees.py
Scratches Zomato Delivery Fees.py Zomato Delivery Fees.py
Project
    VOX Cinema Ticket System.py RTA Driving License Information System.py Zomato Delivery Fees.py
    7     amount =int(input("Enter Total Payable Amount"))
    8
    9     if(amount >100):
    10        print("Congratulation ")
    11        print("You are food will be delivered in 30min")
    12        print("Enjoy our free Delivery Service")
    13        print("Bon Appetite ")
    14        print("For More Contact Us @ +971 524707211 ")
    15
    16    else:
    17        print("Congratulation")
    18        print("You are food will be delivered in 30min")
    19        print("To Enjoy our free Delivery Service")
    20        print("Minimum Purchase Order AED 100")
    21        print("Bon Appetite ")
    22        print("For More Contact Us @ +971 524707211")
    23
    24
Run: Delivery Fees × Zomato Delivery Fees ×
* C:\Users\Suren Raj Tualdhari\PycharmProjects\LO3\venv\Scripts\python.exe "C:\Users\Suren Raj Tualdhari\AppData\Roaming\JetBrains\PyCharmCE2022.1\scratches\Zomato Delivery Fees.py"
File "C:\Users\Suren Raj Tualdhari\AppData\Roaming\JetBrains\PyCharmCE2022.1\scratches\Zomato Delivery Fees.py" Line 14
else
    ^
SyntaxError: invalid syntax
Process finished with exit code 1

```

This example I have taken from my very source code. Here is the part of debugging it is mentioned that the error line i.e. Line number 14 for this example and is highlighted as “else”. Also, as you can see, a syntax error popped up.

- Run-Time Error:** Mostly computer programs crash or get terminated in the mid-way while debugging. Even if the program code seems correct all the syntax is done accordingly. Most of the time nothing is wrong with it but the debugger shows the runtime error. It is likely to occur when something is divided by zero, when there is insufficient memory for dynamic allocation, referencing an out-of-range array element etc. So, when the debugging is executed run time error is likely to pop out. (Mishra, 2022)
- Semantic Error:** Most often logical errors are also known as semantic errors. & this error is often referred to as errors which are not detected by the compiler as a result the output is incorrect. However, the semantic errors in the program hardly affect the execution process as it does not give any output of reporting any errors & after the program is executed. This kind of error occurs due to an incorrect algorithm, lack of proper understanding of the problem, and lack of clarity of the hierarchy of the operator. (Mishra, 2022)

Integrated Development Environment (IDE) Feature:

Integrated development environment IDE is an application used to create or code software. It consists of tools such as a text editor, debugging errors etc. These tools will assist program developers to code software.

- **Text Editor:** It is an important phase of programming as it allows the programmer to write code. an empty canvas for the artist. However, in this canvas, this programmer will use simple text which will be translated into machine code by an interpreter or compiler. (Isaac Computer Science, 2022)

```
def greetings(name):
    print(f'Hello {name}, welcome to my IDE')

if __name__ == '__main__':
    greetings('world')
```

In the above picture, we can see the sample of the text editor.

- **Runtime Environment:** The runtime environment feature of IDE allows developers to run their code by converting the source code file into a machine code file to make it easier for the computer. Once high-level program code is converted into machine language, compiler Programs can also be run in IDE to check whether it is working properly.

```
countdowntimer function.py
```

```
1 from time import sleep
2
3 def countdown_timer(start):
4     while start > 0:
5         print(start)
6         sleep(1)
7         start -=1
8
9 if __name__ == '__main__':
10    countdown_timer(10)
```

```
Running: countdowntimer function.py
```

```
10
9
8
7
6
5
4
3
2
1
>>> |
```

- **Syntax Checking:** Quite similar to spelling correction where errors made are highlighted & indicated by a red underline in sentences. IDE also do these checking like identifying and highlighting syntax errors such feature is available in most IDEs.

!	Line	Description	File	Project	Path
×	15	; expected (CS1002)	Program.cs	PasswordGuess	Program.cs

In the above picture, we can see the syntax checking as highlighted in the line where the error has occurred. For example in this case it's line number 15.

- **Keyword Highlighting:** Also known as colour coding or pretty printing generally denote the colours that an IDE applies automatically when a developer writes the codes.

```

pass_or_fail.js
1 v function passOrFail(testScore) {
2 v   if (testScore >= 60) {
3     return "Pass";
4   }
5 v   else {
6     return "Fail";
7   }
8 }
9
10 alert(passOrFail(50));

```

In this picture, we can observe that JavaScript codes are highlighted. Generally, every colour is highlighted depending on whether the code is variables, commands, functions, parameters or datatypes.

- **Debugging Tools:** In the world of programming, the term is used to denote an error which blocks the program to run successfully. As a result, crashing false results output often a programmer has to deal with. IDE has a dedicated debugging feature which assists to track, bugs. Bugs like syntax errors don't tend to stop the program from executing however, they only surfaced when programs produce incorrect output which crashes the software altogether.
- **Threading:** The programming term for threading refers to the single sequence code which is being executed. When a developer uses multiple threads, it becomes harder to find bugs manually. However, IDE has a thread inspector feature which helps to find the threads that are running & monitor the status of each thread.
- **Version Control:** As the name suggests the version control feature allows the developer to record the changes which have been made to the program over time. As the result, the programmer can record the change every time so it will help to find where the change has been made in the program code. In case the code gets messed up the programmer use this feature as the reference point.

P5

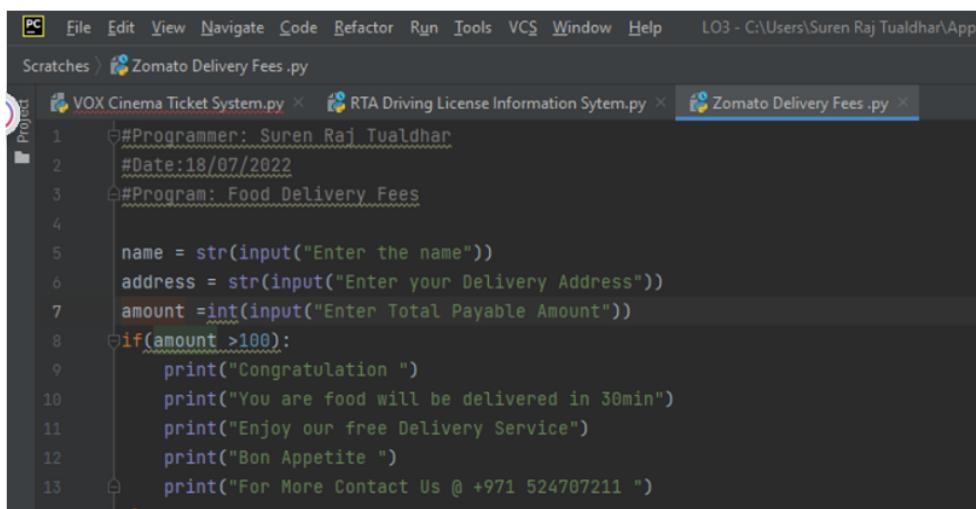
4

Coding Standards

They are a series of procedures that can be defined for a particular programming language specifying a programming style, methods, & different procedures. These procedures can be for various aspects of the program written in that language. They can be considered essential attributes of software development. A coding standard makes sure that all the developers working on the project are following certain specified guidelines. The code can be easily understood, and proper consistency is maintained.

Code Comments & Proper Documentation

For a professional developer mentioning the comments while coding a program is a must practice. It helps successor developers to understand the nature of the program code. A good practice for writing a comment is it should be written at the start of the program. Comment start with the # Symbol. So anything written after the hashtag is not executed by IDE. Documentation of the code is equally important for a program to execute and store properly. With the wrong code, the program will not behave as expected. (PEP 8 – Style Guide for Python Code | peps.python.org, 2022)



```
#Programmer: Suren Raj Tualdhar
#Date:18/07/2022
#Program: Food Delivery Fees

name = str(input("Enter the name"))
address = str(input("Enter your Delivery Address"))
amount =int(input("Enter Total Payable Amount"))

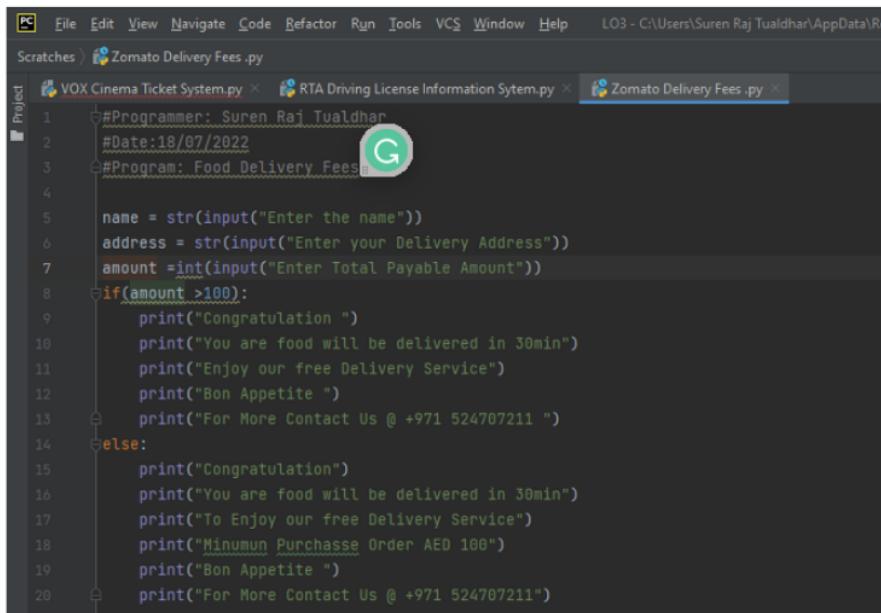
if(amount >100):
    print("Congratulation ")
    print("You are food will be delivered in 30min")
    print("Enjoy our free Delivery Service")
    print("Bon Appetite ")
    print("For More Contact Us @ +971 524707211 ")

else:
```

So in the above picture, we can see the code comments. Which will make the identification of the program easier for any developer.

Use of Indentation

As a professional developer, your work should be organized and kept clean. So if you have to go through the code in the future it will be a lot easier to understand the program. There is no particular style for the indentation for writing a program code. However, throughout the whole program code, only one particular & consistent indentation style should be implemented. (PEP 8 – Style Guide for Python Code | peps.python.org, 2022)



The screenshot shows a PyCharm IDE interface with three tabs open: 'VOX Cinema Ticket System.py', 'RTA Driving License Information System.py', and 'Zomato Delivery Fees.py'. The 'Zomato Delivery Fees.py' tab is active. The code is as follows:

```
1 #Programmer: Suren Raj Tualdhar
2 #Date:18/07/2022
3 #Program: Food Delivery Fees G
4
5 name = str(input("Enter the name"))
6 address = str(input("Enter your Delivery Address"))
7 amount =int(input("Enter Total Payable Amount"))
8 if(amount >100):
9     print("Congratulation ")
10    print("You are food will be delivered in 30min")
11    print("Enjoy our free Delivery Service")
12    print("Bon Appetite ")
13    print("For More Contact Us @ +971 524707211 ")
14 else:
15     print("Congratulation")
16     print("You are food will be delivered in 30min")
17     print("To Enjoy our free Delivery Service")
18     print("Minumun Purchasse Order AED 100")
19     print("Bon Appetite ")
20     print("For More Contact Us @ +971 524707211")
```

Here, from my source code I have used the same indentation throughout the whole program.

Avoid Commenting On Obvious Things

While maintaining the coding standards one should be in mindfulness that not to ever do it. It is a common mistake to add unnecessary comments while writing a program. As the program code can look clustered & clumsy with too much of brackets and comments. As a developer keeping code simple should be the 1st priority clumsy. We can see the sample in the diagram below.

```
for ( int i=0; i<10; i++) //for loop starts here
{
    // starting brace

    // logic starts here

}
//ending brace
```

Naming Conventions

Variables, functions, classes & methods should be meaningful when it comes to the naming conventions. Naming convention in other words means to give a meaningful name which indicates the code & its role with all the variables and functions of the code. There are reserved words for naming variables which should be avoided to make the code much more readable & understandable for the program developers.

(PEP 8 – Style Guide for Python Code | peps.python.org, 2022)

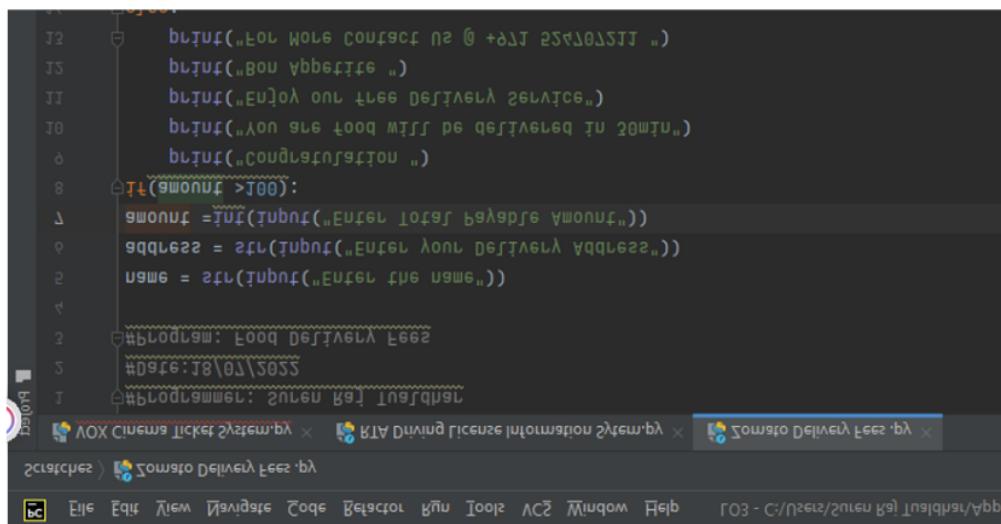
```
#code 1
def func(a):
    x = 0
    if a % 2 == 0:
        x = True
    else:
        x = False
    return x
```

```
#code 2
def checkEvenOdd(num):
    isEven = 0
    if num % 2 == 0:
        isEven = True
    else:
        isEven = False
    return isEven
```

In the above picture, Both code1 and code2 work perfectly and even give the same output but for a better understanding of the program the code2 is more clear to the developer compared to code1

Spaces & Tabs

Spaces and tabs are the most popular way of indentation methods in python. The mixture of both tabs & spaces for Indentation is not recommended in python. As the preferred indentation method is Space whereas Tabs are generally used to maintain consistent indentation



Here, in the above picture, we can see the general use of the spaces and tabs in the simplest form.

Maximum line length

Python has some limitations in its coding standard to keep things simple and understandable. Therefore, while writing the program a developer can write only 79 characters at maximum in the single line length. This restriction helps the program developers to keep things in order and simplified. So in unavoidable circumstances, the line is broken into multiple lines.

```
# Breaking into multiple lines.
num_list = [y for y in range(100)
            if y % 2 == 0
            if y % 5 == 0]
print(num_list)
```

So in the above picture, we can see how a line is broken into multiple lines.

Imports

As a data scientist most often the thing you use in python is the importin of the libraries or modules. This makes the work easier for them as they have to work with data. However, the import is always done at the start of the script. So as the last but not the least coding standard as a developer one should follow the sequence. In multiple imports, a developer or data scientist should mention it in a single line.

The libraries which are imported are grouped following order:

- standard library
- third party imports
- local application or library-specific imports

```
"""
    Algos module consists of all the basic algor
"""

from __future__ import print

__all__ = ['searching', 'sorting']
__version__ = '0.0.1'
__author__ = 'Chitrang Dixit'

import os
import sys
```

For better readable & understandable the developer has to ensure there is a line break or a blank line is included after each group of library imports so that it is. (PEP 8 – Style Guide for Python Code | peps.python.org, 2022)

M4

5

Debugging Process Used To Help Develop More Secure & Robust Applications

Debugging is the process of identifying the different types of errors & bugs. After a good analysis of the problem, the developer should also come up with solutions for fixing the errors & bugs in the program. The data which is collected from debugging process can be used to verify the application is free of error ensuring the quality of the application is achieved along with output results. After the debugging is done the software or the application is once again sent back to the tester to continue the testing process.

```

File Edit View Navigate Code Refactor Run Tools VCS Window Help L03 - C:\Users\Suren Raj Tualdhar\AppData\Roaming\JetBrains\PyCharmCE2022.1\scratches\Driving License Information System.py
Scratches > Driving License Information System.py > Delivery Fees.py
4
5     name = str(input("Enter the name"))
6     age = int(input("Enter your age"))
7     if(age >18):
8         print("You ARE ELIGIBLE FOR USING dl SYSTEM")
9         print("Enter your choice from 1 to 5")
10        print("Choice 1: Opening A File")
11        print("Choice 2: Issuing License")
12        print("Choice 3: Handbook Manual")
13        print("Choice 4: Knowledge and Innovation")
14
15        choice = int(input("Enter your choice"))
16        (choice == 1):
17            print ("Your Opening File fee is 200AED")
18        elif(choice == 2):
19            print("Your License Issuing fee is 600AED")
20        elif(choice == 3):
21            print("Your License Issuing fee is 50AED")
if(age >18)
Run: Delivery Fees > Driving License Information System
C:\Users\Suren Raj Tualdhar\PycharmProjects\L03\venv\Scripts\python.exe "C:\Users\Suren Raj Tualdhar\AppData\Roaming\JetBrains\PyCharmCE2022.1\scratches\Driving License Information System.py" line 16
(choice == 1):
^
SyntaxError: invalid syntax
Process finished with exit code 1
Breakpoint reached (52 minutes ago)
Version Control Run Debug Python Packages TODO Python Console Problems Terminal Services
2:58 CRUFT UTF-8 4 spaces Python 3.9 (L03)

```

Here I have listed some benefits of debugging.

- Immediate reports of the error condition, make it easier to detect the error prior and make the process of program development stress-free and smooth.
- It allows or provides maximum useful information on data structure for the interpretation process to be executed in a much simpler and easier manner.
- It saves time, and energy and makes the developer stress free by helping to avoid complex one-use testing code.

In conclusion, if the software or applications is free of errors & bugs the software will work properly after the debugging

Whereas the testing is done to find such errors and bugs in the software so that the program software can function properly when handed over to the client. Nevertheless, testing cannot guarantee that the software is 100% bug-free. Testing can assist to provide working results for specific testing environments. Testing is the initial phase of debugging so after the debugging process the program software is again tested to check if there are more or new bugs and errors which will hamper the workflow of the software. (Testing Vs Debugging | Professionalqa.com, 2022) =

To summarise the whole picture the bug is an error in a program that hampers the execution process resulting in the program crashes the general true expected output is not achieved from the software. As we have discussed earlier that the IDE has a debugging feature which helps in the debugging process. Debugging tool of IDE assists the program developers to pinpoint and trace the errors like syntax errors, runtime errors etc. This debugging feature helps the developers to find the origin of the errors and bugs. Moreover, the developer can determine the required solution for the problem and work towards solving the error & bug. After the debugging is done the retest of the program software is done so that it can execute properly without any further failures. Thus the process of debugging is the process of eliminating the flaws & glitches in an application in a systematic way so the application function well. Also, it helps the program developer to develop & make the applications free secure & more robust.

The screenshot shows the PyCharm IDE interface. In the top navigation bar, the file 'Zomato Delivery Fees.py' is selected. The code editor displays the following Python script:

```
amount =int(input("Enter Total Payable Amount"))
if(amount >100):
    print("Congratulation ")
    print("You are food will be delivered in 30min")
    print("Enjoy our free Delivery Service")
    print("Bon Appetite ")
    print("For More Contact Us @ +971 524707211")
else:
    print("Congratulation")
    print("You are food will be delivered in 30min")
    print("To Enjoy our free Delivery Service")
    print("Minimum Purchasse Order AED 100")
    print("Bon Appetite ")
    print("For More Contact Us @ +971 524707211")
```

A red box highlights the line 'else:' at line 14. The run tab below shows the command: "C:\Users\Suren Raj Tualdhar\PycharmProjects\L03\venv\Scripts\python.exe" "C:\Users\Suren Raj Tualdhar\AppData\Roaming\JetBrains\PyCharmCE2022.1\scratches\Zomato_Delivery_Fees.py". The status bar indicates 'Line 14'. The terminal output shows a 'SyntaxError: invalid syntax' and 'Process finished with exit code 1'.

In the above diagram, I encountered an error while developing the program for this assignment

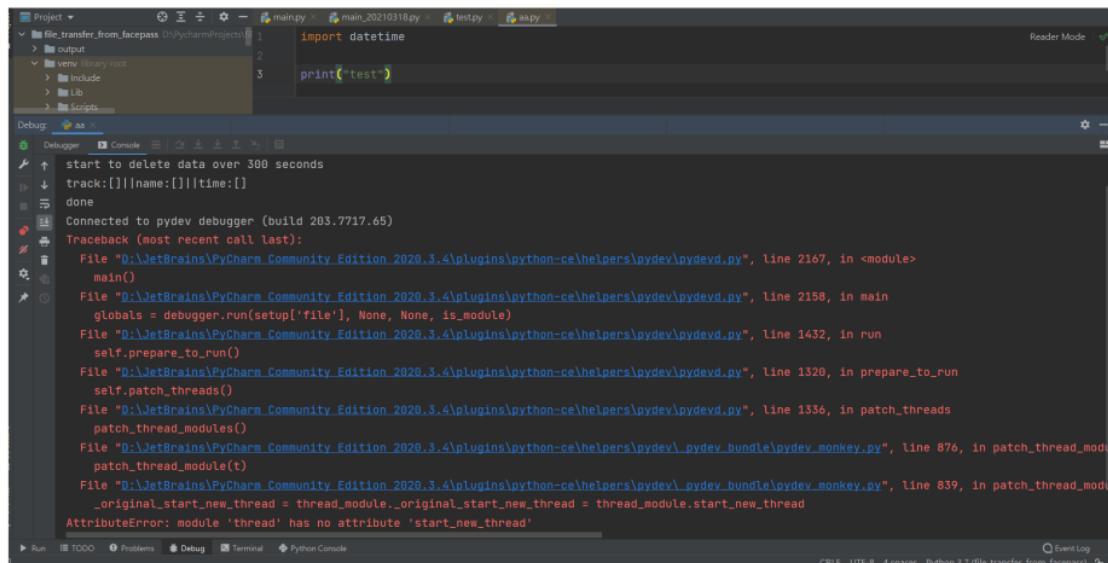
Debugging is not about fixing bugs and reporting back bug reports. Some error also happens during coding as writing code as the result the programs cannot work properly when there are errors & bugs in the source code like syntax errors. Mistakes may happen while writing the program code for example error grammar, and syntax errors which affects the program execution process etc.

Translators are not programmed to execute the type of computer program which has program code error written wrongly. Some errors can cause the program to terminate during the execution process even though the code seems correct such types of errors are generally referred to as runtime errors.

Runtime errors such as invalid data input or division by 0 can lead a program to crash.

In the above diagram, I encountered a syntax error while developing the program for this assignment.

Talking about another error type which is logical errors. It is an error which occurs when even though the compiling and execution process is successful but the results are way too different from what the program was designed to deliver. Hence, debugging comes to help to fix all these different errors even if the application when is used application will function smoothly as it was designed to. (How debugging can make you a better developer | The Man in the Arena, 2022)



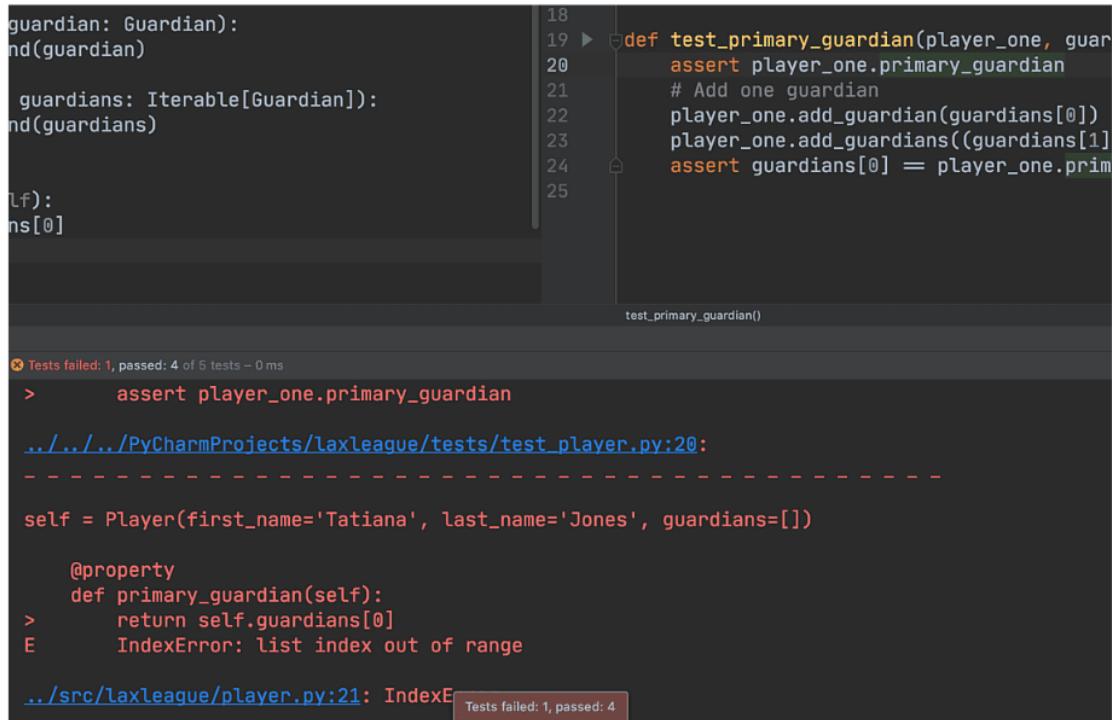
The screenshot shows the PyCharm IDE interface. In the top right corner, there is a green checkmark icon labeled 'Reader Mode'. The main window displays a Python file named 'main.py' with the following code:import datetime
print('test')The code is highlighted in blue. Below the code editor, the 'Console' tab is active, showing the following output:

```
start to delete data over 300 seconds  
track:[]|name:[]|time:[]  
done  
Connected to pydev debugger (build 203.7717.65)  
Traceback (most recent call last):  
File "D:\JetBrains\PyCharm_Community_Edition_2020.3.4\plugins\python-ce\helpers\pydev\pydevd.py", line 2167, in <module>  
main()  
File "D:\JetBrains\PyCharm_Community_Edition_2020.3.4\plugins\python-ce\helpers\pydev\pydevd.py", line 2158, in main  
globals = debugger.run(setup['file'], None, None, is_module)  
File "D:\JetBrains\PyCharm_Community_Edition_2020.3.4\plugins\python-ce\helpers\pydev\pydevd.py", line 1432, in run  
self.prepare_to_run()  
File "D:\JetBrains\PyCharm_Community_Edition_2020.3.4\plugins\python-ce\helpers\pydev\pydevd.py", line 1320, in prepare_to_run  
self.patch_threads()  
File "D:\JetBrains\PyCharm_Community_Edition_2020.3.4\plugins\python-ce\helpers\pydev\pydevd.py", line 1336, in patch_threads  
patch_thread_modules()  
File "D:\JetBrains\PyCharm_Community_Edition_2020.3.4\plugins\python-ce\helpers\pydev\pydev_bundle\pydev_monkey.py", line 876, in patch_thread_modules  
patch_thread_module(t)  
File "D:\JetBrains\PyCharm_Community_Edition_2020.3.4\plugins\python-ce\helpers\pydev\pydev_bundle\pydev_monkey.py", line 839, in patch_thread_modules  
_original_start_new_thread = thread_module._original_start_new_thread = thread_module.start_new_thread  
AttributeError: module 'thread' has no attribute 'start_new_thread'
```

The bottom status bar indicates the file is 'C:\...\file_transfer_from_facesapi.py' and the encoding is 'UTF-8'.

Debugging is an important process of pinpointing and removing the coding errors from part of software development. As a developer, one needs to, use & rely on debugging process as it provides many advantages for them. It saves time for the developers as it can easily identify the bugs and errors which later on can be rectified as soon as they are reported instead of searching them manually searching which is quite stress full and tedious process. As the result easily identification of the error in code makes it easier for a developer to find solutions. The error reports are in a structured & systematic manner, as a result, the errors & bugs are identified very quickly by the developers by undergoing the debugging process they can fix, rectify & correct as per the requirements which it much smoother for the developers to continue their entire development process.

All software & applications can't be bugs & errors free especially during the development process. The process of identifying & fixing the bugs & errors in the software known as the process of debugging becomes very important. As an important process debugging is also an essential phase where developers can solve unwanted errors with bugs in a small amount of time. If it is left unsolved it can lead the system to crash as a result the software development costs will be very expensive. With the help of debugging process program developers can save costs of developing the software but also increase the productivity & efficiency of the process along with the lifecycle of software development.



The screenshot shows a PyCharm IDE interface. On the left is the code editor with Python code related to a 'Player' class and its 'guardians' attribute. On the right is the 'test_player.py' file containing a single test case 'test_primary_guardian'. The test fails with an assert error. Below the code editor is a terminal window showing the test results: 1 test failed, 4 passed, and 0 ms execution time. The error message points to line 20 of the test file, which contains the failing assert statement. The stack trace shows the error originates from line 21 of the player class's primary guardian implementation, where an IndexError is raised due to a list index out of range error.

```
guardian: Guardian):
    nd(guardian)

    guardians: Iterable[Guardian]):
        nd(guardians)

    lf):
        ns[0]

18
19 def test_primary_guardian(player_one, guar
20     assert player_one.primary_guardian
21     # Add one guardian
22     player_one.add_guardian(guardians[0])
23     player_one.add_guardians((guardians[1]
24     assert guardians[0] == player_one.prim
25

test_primary_guardian()

Tests failed: 1, passed: 4 of 5 tests - 0 ms
>     assert player_one.primary_guardian
.../.../.../PyCharmProjects/laxleague/tests/test_player.py:20:
-----
self = Player(first_name='Tatiana', last_name='Jones', guardians=[])
@property
def primary_guardian(self):
>     return self.guardians[0]
E     IndexError: list index out of range
.../src/laxleague/player.py:21: IndexError
Tests failed: 1, passed: 4
```

The above figure is an example of an error during debugging test

The program should be bugs-free & and the error should be fixed at the earliest during the initial stages of the development lifecycle. This will help the developers to improve the quality delivered by the software. Thus debugging is an essential process which helps program developers to fix errors & bugs in their code to make the program work flawless. It is important to develop a program secured, errors free along with robust applications. (What is Debugging? - Definition from Techopedia, 2022)

Reference:

- PROGRAMMING. 2022. *Key Features of Event-Driven Programs..* [online] Available at: <<https://zakkcuthbertunit14eventdrivenprogramming.wordpress.com/2014/11/07/key-features-of-event-driven-programs/>> [Accessed 11 July 2022].
- home, P., support, c. and </form>, 2022. *What Is an Object-Oriented Database {Concepts, Examples, Pros and Cons}?* [online] Knowledge Base by phoenixNAP. Available at: <<https://phoenixnap.com/kb/object-oriented-database>> [Accessed 12 July 2022].
- InterviewBit. 2022. *Top Characteristics of Object Oriented Programming.* [online] Available at: <<https://www.interviewbit.com/blog/characteristics-of-object-oriented-programming/>> [Accessed 12 July 2022].
- Slideplayer.com. 2022. *CSC238 INTRODUCTION TO OOP Chapter ppt video online download.* [online] Available at: <<https://slideplayer.com/slide/5788118/>> [Accessed 12 July 2022]
- StudyBay. 2022. *Event-Driven Programming - Applications & Features.* [online] Available at: <<https://studybay.com/blog/event-driven-development-features/>> [Accessed 13 July 2022].
- Ukeessays.com. 2022. *Event-Driven Programming Features.* [online] Available at: <<https://www.ukessays.com/essays/computer-science/eventdriven-programming-features-6167.php>> [Accessed 13 July 2022].
- Isaac Computer Science. 2022. *Isaac Computer Science.* [online] Available at: <https://isaaccomputerscience.org/concepts/prog_softeng_ide?examBoard=all&stage=all> [Accessed 3 August 2022].
- Mishra, N., 2022. *Types of Errors in Programming.* [online] The Crazy Programmer. Available at: <<https://www.thecrazyprogrammer.com/2014/08/types-of-errors-in-programming.html>> [Accessed 3 August 2022].
- Peps.python.org. 2022. *PEP 8 – Style Guide for Python Code | peps.python.org.* [online] Available at: <<https://peps.python.org/pep-0008/>> [Accessed 3 August 2022].
- 2022. [online] Available at: <<https://www.datacamp.com/tutorial/pep8-tutorial-python-code>> [Accessed 3 August 2022].
- Pymbook.readthedocs.io. 2022. *PEP8 Guidelines — Python for you and me 0.5.beta1 documentation.* [online] Available at: <<https://pymbook.readthedocs.io/en/latest/pep8.html#code-lay-out>> [Accessed 3 August 2022].

- Python, R., 2022. *How to Write Beautiful Python Code With PEP 8 – Real Python*. [online] Realpython.com. Available at: <<https://realpython.com/python-pep8/>> [Accessed 3 August 2022].
- Professionalqa.com. 2022. *Testing Vs Debugging* |Professionalqa.com. [online] Available at: <<https://www.professionalqa.com/testing-vs-debugging>> [Accessed 3 August 2022].
- The Man in the Arena. 2022. *How debugging can make you a better developer* | *The Man in the Arena*. [online] Available at: <<https://carlalexander.ca/how-debugging-make-better-developer/>> [Accessed 3 August 2022].

4.

TA_Submission_Suren

ORIGINALITY REPORT



PRIMARY SOURCES

1	essayabortion.blogspot.com Internet Source	2%
2	documents.mx Internet Source	1%
3	WWW.coursehero.com Internet Source	1%
4	Submitted to The European College of Business and Management (ECBM) Student Paper	1%
5	Submitted to UK College of Business and Computing Student Paper	1%
6	Submitted to The University of the South Pacific Student Paper	1%
7	Submitted to The Hong Kong Institute of Education Student Paper	1%
8	bootpoot.tech Internet Source	1%

9	Submitted to City of Sunderland College, Sunderland Student Paper	1 %
10	Submitted to Walsall College Student Paper	1 %
11	Submitted to Barking College Student Paper	<1 %
12	Submitted to Institute of Technology & Management, Ugep Student Paper	<1 %
13	Submitted to American InterContinental University Student Paper	<1 %
14	Submitted to Sim University Student Paper	<1 %
15	intellipaat.com Internet Source	<1 %
16	Submitted to Heart of Worcestershire College Student Paper	<1 %
17	bigthink.com Internet Source	<1 %
18	rosettacode.org Internet Source	<1 %
19	codereview.stackexchange.com Internet Source	<1 %

Exclude quotes Off

Exclude bibliography On

Exclude matches Off