iCAT
Design & Media College

BONAFIDE CERTIFICATE

This is to certify that this record of course work is a bonafide work done by **Ajil Padathuparambil Pappachan**, ID No.: **2018UG03077**, in partial fulfillment of requirements for the **1st year B.Sc. Game Programming** during the academic year 2018 – 2019 and is the original work of the candidate.

Submitted for the **Programming and Interactive Media** module assessment held on _____________.

_______________ _______________

Verified By **Staff In-Charge**

RESEARCH ON C++

BY AJIL PAPPACHAN (L4 GDV)

INTRODUCTION

C++ is a programmer's best friend when Coding and Software design is considered. Although it has been defined in many ways over various sources, they all basically say the same thing.

C++ (pronounced see-plus-plus) was developed by Bjarne Stroustrup, as an extension of the C language. It is an intermediate level, general-purpose object-oriented programming (OOP) language, having the features of both high and low-level languages. Initially named "C with classes" (because it had all the properties of C with the addition of "Class" concept), it was renamed in 1983 to C++. C++ can be used to code in "C style" or "Object Oriented style". Therefore, it is an example of a hybrid language.

(Google.co.in, 2018)

C++ is one of the most popular IDE for making System Software, Drivers, Client-Server Applications, Embedded Firmware, etc,

C++ has a huge collection of predefined classes (Data types that can be instantiated multiple times) and facilitates the declaration of user-defined classes. Classes consist of member functions which have some specific functionality. One class can have multiple Objects which can be defined to implement the functions inside the class. Objects can be defined as "Instances of a class which are created at runtime). The properties of one class can also be inherited by another class.

C++ includes Operators for performing Comparison, arithmetic, bit manipulation, and logical operations. C++ also enables the overloading of operators (eg: addition).

The most important concepts of C++ are Polymorphism, Virtual and Friend functions, Namespaces, Templates, Pointers, etc.

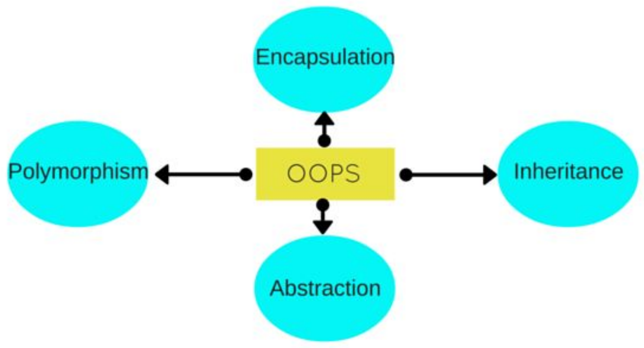
DIFFERENCE BETWEEN C AND C++

- C was released in 1972, C++ was first released in 1983.
- C is a Procedural Computing Language, C++ is an Object Oriented Programming Language. i.e, C++ contains Classes and Objects, which provide new ways to structure code.
- C++ provides solutions such as encapsulation, namespaces for variables, and improved error-handling processes. It allows for object reuse and other various manipulations of the object as a data item.
- C++ builds on the procedural C language by adding the functionalities that represent the object-oriented programming philosophy.

OBJECT ORIENTED PROGRAMMING (OOP) CONCEPTS

Object-oriented programming (OOP) is a programming model based on objects. In this model, Data is compartmentalized into Objects (Data Fields) and Object contents and behavior are described through the declaration of Classes (Methods).

Object-Oriented Programming style is based on the concept of Class, Objects, and other concepts related to them (Inheritance, Polymorphism, Abstraction, Encapsulation, etc.)



(Studytonight.com, 2018)

Object-oriented programming makes programming easier. It provides reusability, refactoring, extensibility, maintenance, and efficiency.

OOP’S modular design lets programmers make software in manageable blocks of code instead of large amounts of sequential code. It has been, therefore, the Programming model of choice for the last decade.

OOP provides scalability. i.e, Objects and definitions have no finite limitations. Also, unlike in old linear languages, where a bug in the linear code would result in masses of untraceable errors, OOP languages separate method and data and makes error checking easier.

<u>**OOPS CONCEPTS DEFINITION**</u>

1. Objects
2. Classes
3. Abstraction
4. Encapsulation
5. Inheritance
6. Overloading
7. Exception Handling

Objects
Objects are the basic unit of OOP. They are instances of a class, which have data members and uses various member functions to perform tasks.

Class
It is similar to structures in C language. A class can also be defined as user-defined data type but it also contains functions in it. So, a class is basically a blueprint for an object. It declares & defines what data variables the object will have and what operations can be performed on the class's object.

Abstraction
Abstraction refers to showing only the essential features of the application and hiding the details. In C++, classes can provide methods to the outside world to access & use the data variables, keeping the variables hidden from direct access, or classes can even declare everything accessible to everyone, or maybe just to the classes inheriting it. This can be done using access specifiers.

Encapsulation
It can also be said data binding. Encapsulation is all about binding the data variables and functions together in class.

Inheritance

Inheritance is a way to reuse once written code again and again. The class which is inherited is called the **Base** class & the class which inherits is called the **Derived** class. They are also called parent and child class.

So when a derived class inherits a base class, the derived class can use all the functions which are defined in the base class, hence making code reusable.

Polymorphism

It is a feature, which lets us create functions with same name but different arguments, which will perform different actions. That means, functions with the same name, but functioning in different ways. Or, it also allows us to redefine a function to provide it with a completely new definition. You will learn how to do this in details soon in coming lessons.

Exception Handling

Exception handling is a feature of OOP, to handle unresolved exceptions or errors produced at runtime.

(Studytonight.com, 2018)

Syntax and Structure of C++ program

Header files are included at the beginning just like in C program. Here iostream is a header file which provides us with input & output streams. Header files contained predeclared function libraries, which can be used by users for their ease.

Using namespace std tells the compiler to use the standard namespace. Namespace collects identifiers used for class, object, and variables. A namespace can be used by two ways in a program, either by the use of using statement at the beginning, as we did in above-mentioned program or by using the name of a namespace as a prefix before the identifier with scope resolution (::) operator.

Example : std::cout << "A";

main(), is the function which holds the executing part of the program its return type is int.

cout <<, is used to print anything on the screen, same as printf in C language. **cin** and **cout** are same as scanf and printf, the only difference is that you do not need to mention format specifiers like %d for int etc, in cout & cin.

Comments

For single line comments, use // before mentioning comment, like

cout<<"single line"; // This is single line comment. For multiple line comment, enclose the comment between /* and */ /*this is a multiple line comment */

Using Classes

Classes name must start with capital letter, and they contain data variables and member functions. This is a mere introduction to classes, we will discuss classes in detail throughout the C++ tutorial.

class Abc{ int i; //data variable void display() //Member Function { cout<<"Inside Member Function"; }}; // Class ends here int main(){ Abc obj; // Creatnig Abc class's object obj.display(); //Calling member function using class object}

This is how class is defined, its object is created and the member functions are used.

Variables can be declared anywhere in the entire program but must be declared, before they are used. Hence, we don't need to declare a variable at the start of the program.

(Studytonight.com, 2018)

<u>Data Types in C++</u>

They are used to define the type of variables and contents used. Data types define the way you use storage in the programs you write. Data types can be built in or abstract.

Built-in Data Types
These are the data types which are predefined and are wired directly into the compiler. eg: int, char etc.

User-defined or Abstract data types
These are the type, that user creates as a class. In C++ these are classes in C it was implemented by structures.

Basic Built in types

char	for character storage (1 byte)
int	for integral number (2 bytes)
float	single precision floating point (4 bytes)
double	double precision floating point numbers (8 bytes)

Example :

char a = 'A'; // character type int a = 1; // integer type float a = 3.14159; // floating point type double a = 6e-4; // double type (e is for exponential)

Other Built-in types

bool	Boolean (True or False)
void	Without any Value
wchar_t	Wide Character

Enum as Datatype

Enumerated type declares a new type-name and a sequence of value containing identifiers which have values starting from 0 and incrementing by 1 every time.

For Example :

enum day(mon, tues, wed, thurs, fri) d;

Here an enumeration of days is defined with variable d. *mon* will hold value 0, *tue* will have 1 and so on. We can also explicitly assign values, like, enum day(mon, tue=7, wed);. Here, *mon* will be 0, *tue* is assigned 7, so *wed* will have value 8.

Modifiers

Specifiers modify the meanings of the predefined built-in data types and expand them to a much larger set. There are four data type modifiers in C++, they are :

8. long
9. short
10. signed
11. unsigned

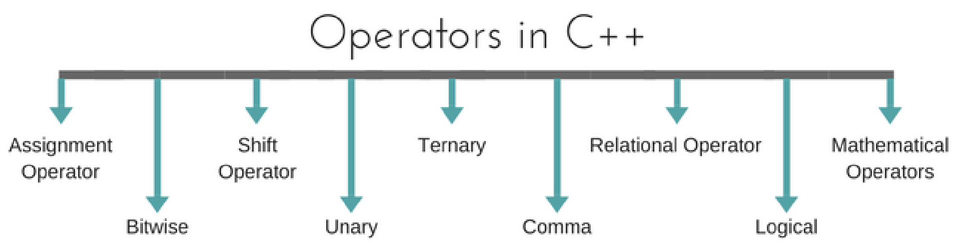
Below mentioned are some important points you must know about the modifiers,

- **long** and **short** modify the maximum and minimum values that a data type will hold.
- A plain int must have a minimum size of **short**.
- Size hierarchy : short int < int < long int
- Size hierarchy for floating point numbers is: float < double < long double
- the **long float** is not a legal type and there are no **short floating point** numbers.
- **Signed** types include both positive and negative numbers and are the default type.
- **Unsigned**, numbers are always without any sign, that is always positive.

(Studytonight.com, 2018)

Operators in C++

Operators are a special type of functions, that takes one or more arguments and produces a new value. For example : addition (+), subtraction (-), multiplication (*) etc, are all operators. Operators are used to performing various operations on variables and constants.



(Studytonight.com, 2018)

Types of operators

12. Assignment Operator

13. Mathematical Operators

14. Relational Operators

15. Logical Operators

16. Bitwise Operators

17. Shift Operators

18. Unary Operators

19. Ternary Operator

20. Comma Operator

Assignment Operator (=)

Operates '=' is used for assignment, it takes the right-hand side (called rvalue) and copy it into the left-hand side (called lvalue). The assignment operator is the only operator which can be overloaded but cannot be inherited.

Mathematical Operators

There are operators used to perform basic mathematical operations. Addition (+) , subtraction (-) , diversion (/) multiplication (*) and modulus (%) are the basic mathematical operators. Modulus operator cannot be used with floating-point numbers.

C++ and C also use a shorthand notation to perform an operation and assignment at the same type.

Example,

int x=10;x += 4 // will add 4 to 10, and hence assign 14 to X. x -= 5 // will subtract 5 from 10 and assign 5 to x.

Relational Operators

These operators establish a relationship between operands. The relational operators are: less than (<), grater than (>), less than or equal to (<=), greater than equal to (>=), equivalent (==) and not equivalent (!=).

You must notice that the assignment operator is (=) and there is a relational operator, for equivalent (==). These two are different from each other, the assignment operator assigns the value to any variable, whereas an equivalent operator is used to comparing values, like in if-else conditions,

Example

int x = 10; //assignment operator x=5;   // again assignment operator if(x == 5)   // here we have used equivalent relational operator, for comparison{ cout <<"Successfully compared";}

Logical Operators

The logical operators are AND (&&) and OR (||). They are used to combine two different expressions together.

If two statements are connected using AND operator, the validity of both statements will be considered, but if they are connected using OR operator, then either one of them must be valid. These operators are mostly used in loops (especially while loop) and in Decision making.

Bitwise Operators

There are used to change individual bits into a number. They work with only integral data types like char, int and long and not with floating point values.

- Bitwise AND operators &
- Bitwise OR operator |
- And bitwise XOR operator ^
- And, bitwise NOT operator ~

They can be used as shorthand notation too, & = , |= , ^= , ~= etc.

Shift Operators

Shift Operators are used to shifting Bits of any variable. It is of three types,

21. Left Shift Operator <<

22. Right Shift Operator >>

Unsigned Right Shift Operator >>>

Unary Operators

These are the operators who work on only one operand. There are many unary operators, but increment ++ and decrement -- operators are most used.

Other Unary Operators: address of &, dereference *, **new** and **delete**, bitwise not ~, logical not !, unary minus - and unary plus +.

Ternary Operator
The ternary if-else ? : is an operator which has three operands.

int a = 10;a > 5 ? cout << "true" : cout << "false"
Comma Operator

This is used to separate variable names and to separate expressions. In the case of expressions, the value of the last expression is produced and used.

Example :

int a,b,c; // variables declaration using comma operator a = b++, c++; // a = c++ will be done.

(Studytonight.com, 2018)

<u>Functions in C++</u>

Functions are used to provide modularity to a program. Creating an application using function makes it easier to understand, edit, check errors etc.

Syntax of Function

*return-type **function-name** (parameters){ // function-body}*

- **return-type:** suggests what the function will return. It can be int, char, some pointer or even a class object. There can be functions which do not return anything, they are mentioned with a **void**.
- **Function Name:** is the name of the function, using the function name it is called.
- **Parameters :** are variables to hold values of arguments passed while function is called. A function may or may not contain parameter list.void sum(int x, int y){ int z; z = x + y; cout << z;}int main(){ int a = 10; int b = 20; **sum (a, b)**;}Here, **a** and **b** are sent as arguments, and **x** and **y** are parameters which will hold values of a and b to perform required operation inside function.
- **Function body:** is the part where the code statements are written.

Declaring, Defining and Calling Function

Function declaration, is done to tell the compiler about the existence of the function. Function's return type, its name & parameter list is mentioned. Function body is written in its definition.

```
#include < iostream>using namespace std;int sum (int x, int y);   //declaring function int main(){
int a = 10; int b = 20; int c = sum (a, b);   //calling function cout << c;}int sum (int x, int y)
//defining function{ return (X + y);}
```

Here, initially the function is **declared**, without body. Then inside main() function it is **called**, as the function returns summation of two values, hence z is their to store the value of sum. Then, at last, function is **defined**, where the body of function is mentioned. We can also, declare & define the function together, but then it should be done before it is called.

Calling a Function

Functions are called by their names. If the function is without argument, it can be called directly using its name. But for functions with arguments, we have two ways to call them,

23. Call by Value
24. Call by Reference

Call by Value

In this calling technique we pass the values of arguments which are stored or copied into the formal parameters of functions. Hence, the original values are unchanged only the parameters inside function changes.

```
void calc(int x);int main(){ int x = 10; calc(x); printf("%d", x);}void calc(int x){ x = x + 10 ;}
```

Output : 10

In this case the actual variable x is not changed, because we pass argument by value, hence a copy of x is passed, which is changed, and that copied value is destroyed as the function ends(goes out of scope). So the variable **x** inside main() still has a value 10.

But we can change this program to modify the original **x**, by making the function **calc()** return a value, and storing that value in x.

```
int calc(int x);int main(){ int x = 10; x = calc(x); printf("%d", x);}int calc(int x){ x = x + 10 ;
return x;}Output : 20
```

Call by Reference

In this we pass the address of the variable as arguments. In this case the formal parameter can be taken as a reference or a pointer, in both the case they will change the values of the original variable.

```
void calc(int *p);int main(){ int x = 10; calc(&x);      // passing address of x as argument
printf("%d", x);}void calc(int *p){ *p = *p + 10;}
```

Output : 20

(Studytonight.com, 2018)

CONCLUSION

There is no denying the fact there are much simpler and capable Software out there that could potentially replace C++. But as of today, C++ is one of the most used programming languages and there are no signs showing it could change anytime soon.

C++ is very efficient at what it does. And some even argue it is the best at what it does. But one thing is for sure. C++ is here to stay.

RESEARCH ON TEXT BASED GAMES

INTRODUCTION

A **text game** or **text-based game** is a video game that uses text characters instead of bitmap or vector graphics.

Text games are typically easy to write and require less processing power than games with graphics, and thus, were more common from 1970 to 1990. However, terminal emulators are still in use today, and people continue playing MUDs (multi-user dungeon) and exploring interactive fiction. Many beginning programmers still create these types of games to familiarize themselves with a programming language, and contests even now are held on who can finish programming a roguelike within a short time period, such as seven days.

While many of the earliest computer games (Adventure, Zork) relied on language parsing due to the command-line-driven, teletype-terminal mainframe environments in which they were developed, the phrase "text-based" is taken to refer not to the user input (though generally keyboard-based) but rather to exclusive use of the fixed-width character display mode, an important distinction to maintain in light of cursor based games such as Rogue and their successors, which employed the characters in the text mode as map symbols rather than as parts of words. Despite enormous differences in display and user interface, the text adventure games and roguelikes both make exclusive use of the text mode and hence are both to be considered text-based.

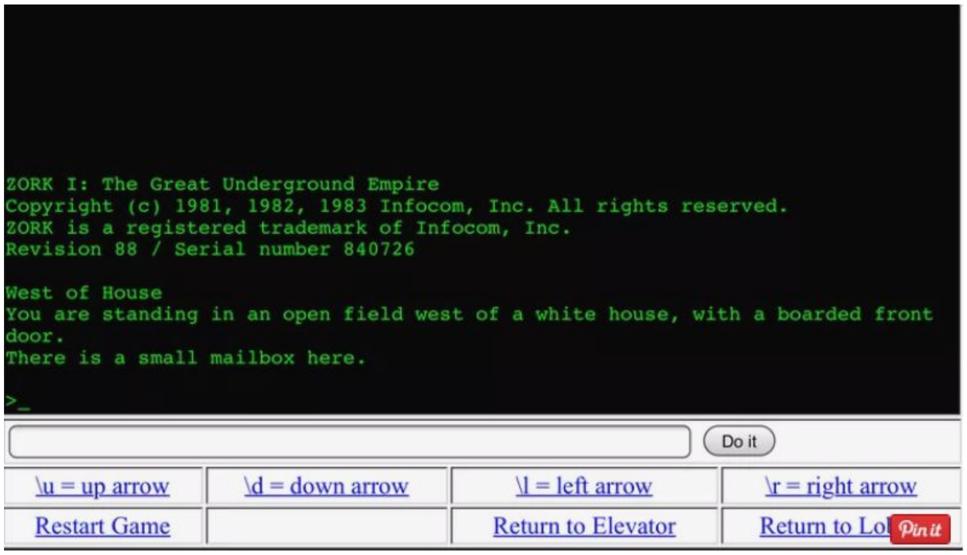
Though punctuation and the alphanumeric symbols can be considered standard in most text modes, many of them contain additional symbols and variant attributes (colours, blinking, lines / columns per screen, etc.) that differ between operating environments: the text mode of a Commodore 64 would be substantially different from that of an IBM PC, though, despite an absence of standardization in the text display (until implementation of later text mode terminal display standards such as VT100 and ANSI), they would both be considered to be text modes. These later standards also contain numerous characters, mostly blocks and lines, specifically intended to be used for fast, low-bandwidth display of crude block graphics in the text mode.

Popular text-based adventure games include Zork and The Hitchhiker's Guide to the Galaxy. (En.wikipedia.org, 2018)

FAMOUS TEXT BASED GAMES

There is no denying the fact that the genre of Text-Based Games is at the risk of going extinct. But it plays a vital role in the history of the Gaming Industry. And it showcases some of the gems of the market. Some famous text-based games are as follows:

ZORK



```
ZORK I: The Great Underground Empire
Copyright (c) 1981, 1982, 1983 Infocom, Inc. All rights reserved.
ZORK is a registered trademark of Infocom, Inc.
Revision 88 / Serial number 840726

West of House
You are standing in an open field west of a white house, with a boarded front
door.
There is a small mailbox here.

>_
```

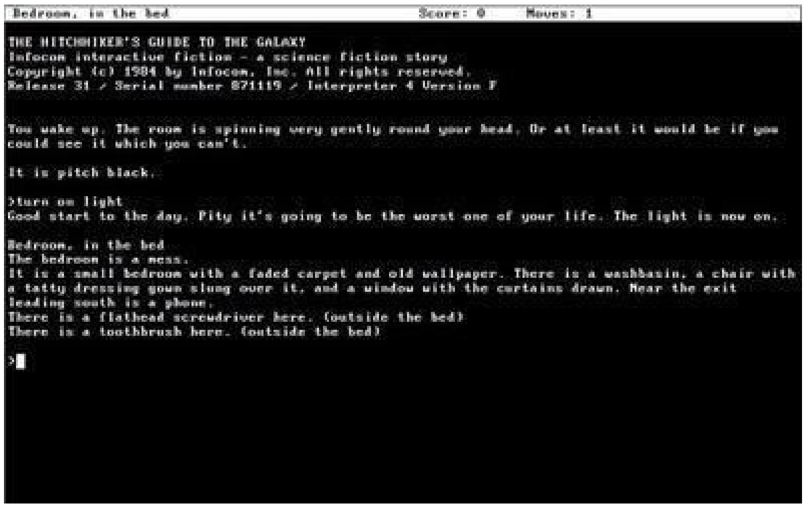
(Lifewire.com, 2018)

Although it was written in the late 1970s, Zork has stood the test of time when it comes to its adventurous storyline. As you traverse through the dungeons in the Great Underground Empire you'll encounter strange creatures, solve tough puzzles and gather up as much loot as you can, armed with nothing but textual descriptions and a command prompt.

One of the text-based genre's shining stars, Zork drops you in an open field next to a white house with a boarded front door and a mailbox. Your escapade begins here, with the next move at your fingertips.

(Lifewire.com, 2018)

The Hitchhiker's Guide to the Galaxy



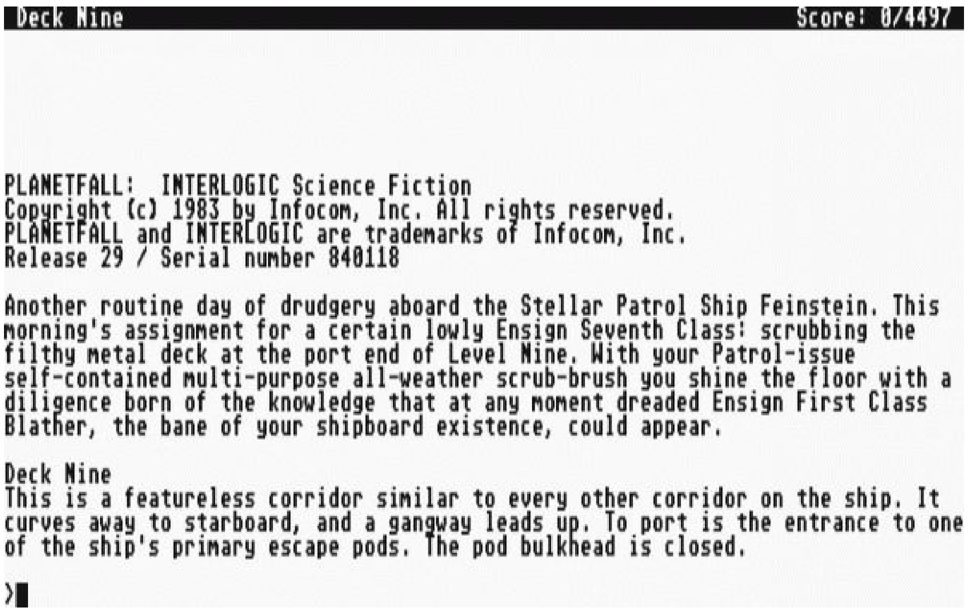
```
 Bedroom, in the bed                              Score: 0      Moves: 1

THE HITCHHIKER'S GUIDE TO THE GALAXY
Infocom interactive fiction - a science fiction story
Copyright (c) 1984 by Infocom, Inc. All rights reserved.
Release 31 / Serial number 871119 / Interpreter 4 Version F


You wake up. The room is spinning very gently round your head. Or at least it would be if you
could see it which you can't.

It is pitch black.

>turn on light
Good start to the day. Pity it's going to be the worst one of your life. The light is now on.

Bedroom, in the bed
The bedroom is a mess.
It is a small bedroom with a faded carpet and old wallpaper. There is a washbasin, a chair with
a tatty dressing gown slung over it, and a window with the curtains drawn. Near the exit
leading south is a phone.
There is a flathead screwdriver here. (outside the bed)
There is a toothbrush here. (outside the bed)

>
```

(Upload.wikimedia.org, 2018)

The Hitchhiker's Guide is a text adventure game, where the player, in the role of Arthur Dent, solves a number of puzzles to complete various objectives to win the game. This includes collecting and using a number of items within their inventory. The player has a limited variety of commands that they can enter to observe, move about, and interact with the game's world, such as "look", "inventory", "north" (to move north) "take screwdriver", or "put robe on hook". Most commands will advance the game's turn counter, and some puzzles require the player to complete the puzzle within a fixed number of turns or else may end the game and require the player to restart at the beginning or a saved state; passive commands like "look" and "inventory", and mistyped or non-comprehended commands do not count as turns. Once the player can acquire it, the player can use the eponymous *Hitchhiker's Guide to the Galaxy* to ask about a wide variety of topics, some which may be helpful in solving the game's puzzles.

(En.wikipedia.org, 2018)

Planetfall



```
Deck Nine                                                Score: 0/4497

PLANETFALL:  INTERLOGIC Science Fiction
Copyright (c) 1983 by Infocom, Inc. All rights reserved.
PLANETFALL and INTERLOGIC are trademarks of Infocom, Inc.
Release 29 / Serial number 840118

Another routine day of drudgery aboard the Stellar Patrol Ship Feinstein. This
morning's assignment for a certain lowly Ensign Seventh Class: scrubbing the
filthy metal deck at the port end of Level Nine. With your Patrol-issue
self-contained multi-purpose all-weather scrub-brush you shine the floor with a
diligence born of the knowledge that at any moment dreaded Ensign First Class
Blather, the bane of your shipboard existence, could appear.

Deck Nine
This is a featureless corridor similar to every other corridor on the ship. It
curves away to starboard, and a gangway leads up. To port is the entrance to one
of the ship's primary escape pods. The pod bulkhead is closed.

>
```

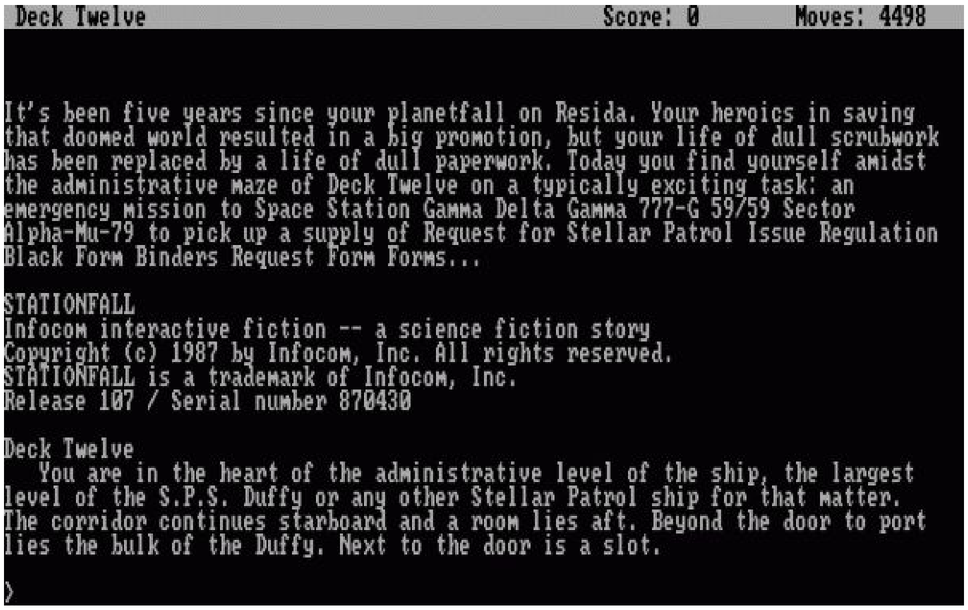
(Ever, 2018)

The Planetfall story begins just after you have transferred to the Spaceship Feinstein where you’re superior, Ensign First Class Blather, is making your life miserable. You have been assigned the rank of Ensign 7th Class and your most important duties are that of a custodian. Obviously, this is not why you enlisted in the Stellar Patrol. Just as you are contemplating going absent without leave your fortunes take a dramatic turn and you find yourself in a situation that just may define the rest of your Stellar Patrol career.

It came out in 1983 and is the perfect game to introduce you into what the 1980s were like for a lot of gamers back then. It might not have the graphics of GTA V, but the story is just as compelling and does an excellent job of drawing you in.

(Riven, 2018)

Stationfall



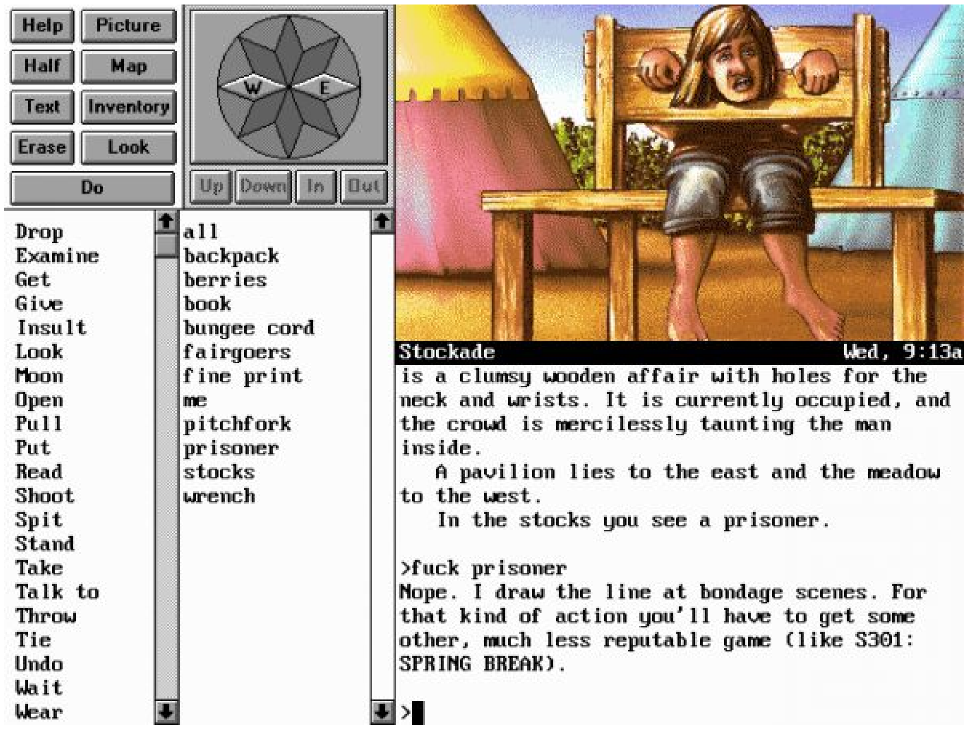
```
Deck Twelve                                         Score: 0        Moves: 4498


It's been five years since your planetfall on Resida. Your heroics in saving
that doomed world resulted in a big promotion, but your life of dull scrubwork
has been replaced by a life of dull paperwork. Today you find yourself amidst
the administrative maze of Deck Twelve on a typically exciting task: an
emergency mission to Space Station Gamma Delta Gamma 777-G 59/59 Sector
Alpha-Mu-79 to pick up a supply of Request for Stellar Patrol Issue Regulation
Black Form Binders Request Form Forms...

STATIONFALL
Infocom interactive fiction -- a science fiction story
Copyright (c) 1987 by Infocom, Inc. All rights reserved.
STATIONFALL is a trademark of Infocom, Inc.
Release 107 / Serial number 870430

Deck Twelve
   You are in the heart of the administrative level of the ship, the largest
level of the S.P.S. Duffy or any other Stellar Patrol ship for that matter.
The corridor continues starboard and a room lies aft. Beyond the door to port
lies the bulk of the Duffy. Next to the door is a slot.

>
```

(Walyou, 2018)

Stationfall is the sequel to *Planetfall*. It has the same main characters, the same type of puzzles and the same type of setting, with impending doom waiting around every corner. It's possible to play it without going through the first one, but it does take out some of the fun because *Planetfall* does an amazing job of developing the characters.

(Walyou, 2018)

Eric the Unready



```
is a clumsy wooden affair with holes for the
neck and wrists. It is currently occupied, and
the crowd is mercilessly taunting the man
inside.
   A pavilion lies to the east and the meadow
to the west.
   In the stocks you see a prisoner.

>fuck prisoner
Nope. I draw the line at bondage scenes. For
that kind of action you'll have to get some
other, much less reputable game (like S301:
SPRING BREAK).

>
```

(Walyou.com, 2018)

So, what's so special about Eric the Unready? It takes a terrible knight, maybe the worst one ever, that's off to save the only person (a princess) that believes in him. It's a game that surprises you at every corner and while seemingly difficult at first is very easy and simple to sift through. It's funny. It doesn't take itself too seriously. It's one of a kind, still after all these years.

(Walyou.com, 2018)

CONCLUSION

Even today, some developers develop new and innovative Text-based games and they are a breath of fresh air in the modern age of high-end graphical video games.

It is an undeniable fact that Text Based Games are almost at the brink of extinction. However, once upon a time, they were the pinnacle of the Gaming industry and the contribution they made to the market will never be forgotten.

BIBLIOGRAPHY

Techopedia.com. (2018). *What is the C++ Programming Language? - Definition from Techopedia*. [online] Available at: https://www.techopedia.com/definition/26184/c-programming-language [Accessed 16 Sep. 2018].

Google.co.in. (2018). *Redirect Notice*. [online] Available at: https://www.google.co.in/url?sa=i&rct=j&q=&esrc=s&source=images&cd=&cad=rja&uact=8&ved=2ahUKEwiJxvWOm7_dAhWJN48KHfVxCB8QjRx6BAgBEAU&url=https%3A%2F%2Fen.wikipedia.org%2Fwiki%2FC%252B%252B&psig=AOvVaw1WpbfYHFPzEJXgd0EFRb3B&ust=1537176537099600 [Accessed 16 Sep. 2018].

Techopedia.com. (2018). *What is the difference between C and C++?*. [online] Available at: https://www.techopedia.com/7/32130/development/what-is-the-difference-between-c-and-c [Accessed 16 Sep. 2018].

Studytonight.com. (2018). [online] Available at: https://www.studytonight.com/cpp/images/oops-concept-basic-cpp.png [Accessed 16 Sep. 2018].

Studytonight.com. (2018). *Object Oriented programming Concepts in C++ | Studytonight*. [online] Available at: https://www.studytonight.com/cpp/cpp-and-oops-concepts.php [Accessed 16 Sep. 2018].

Studytonight.com. (2018). *Basics of C++ Language | C++ Tutorial | Studytonight*. [online] Available at: https://www.studytonight.com/cpp/basics-of-cpp.php [Accessed 16 Sep. 2018].

Studytonight.com. (2018). [online] Available at: https://www.studytonight.com/cpp/images/operators-in-cpp.png [Accessed 16 Sep. 2018].

Studytonight.com. (2018). *Types of operators in C++ | C++ Tutorial | Studytonight*. [online] Available at: https://www.studytonight.com/cpp/operators-and-their-types.php [Accessed 16 Sep. 2018].

Studytonight.com. (2018). *Functions in C++ - Declaration, Definition and Call | C++ Tutorial | Studytonight*. [online] Available at: https://www.studytonight.com/cpp/functions-in-cpp [Accessed 16 Sep. 2018].

En.wikipedia.org. (2018). *Text-based game*. [online] Available at: https://en.wikipedia.org/wiki/Text-based_game [Accessed 16 Sep. 2018].

Lifewire.com. (2018). [online] Available at: https://www.lifewire.com/thmb/xT1N84hCrxSVqJulDdS61oiClaI=/768x0/filters:no_upscale():max_bytes(150000):strip_icc():format(webp)/zork-59ceb4a8af5d3a00114904f5.PNG [Accessed 16 Sep. 2018].

Lifewire. (2018). *The 5 Best Text-Based Video Games*. [online] Available at: https://www.lifewire.com/best-text-based-video-games-4152013 [Accessed 16 Sep. 2018].

Upload.wikimedia.org. (2018). [online] Available at: https://upload.wikimedia.org/wikipedia/en/c/c9/The_Hitchhiker%27s_Guide_to_the_Galaxy_%28video_game%29_screenshot.jpg [Accessed 16 Sep. 2018].

En.wikipedia.org. (2018). *The Hitchhiker's Guide to the Galaxy (video game)*. [online] Available at:

https://en.wikipedia.org/wiki/The_Hitchhiker%27s_Guide_to_the_Galaxy_(video_game) [Accessed 16 Sep. 2018].

Ever, 7. (2018). *Planetfall | Walyou*. [online] Walyou.com. Available at: http://walyou.com/best-text-based-video-games/planetfall/ [Accessed 16 Sep. 2018].

Riven, E. (2018). *7 Greatest Text-Based Adventure Games Ever | Walyou*. [online] Walyou. Available at: http://walyou.com/best-text-based-video-games/ [Accessed 16 Sep. 2018].

Walyou. (2018). *Walyou*. [online] Available at: http://walyou.com/wp-content/uploads//2015/02/Stationfall-e1424700923514@2x.gif [Accessed 16 Sep. 2018].

Walyou.com. (2018). [online] Available at: http://walyou.com/wp-content/uploads//2015/02/Eric-the-unready-e1424701890786.png [Accessed 16 Sep. 2018].