



**NEW HORIZON
COLLEGE OF ENGINEERING**

Autonomous College Permanently Affiliated to VTU, Approved by AICTE & UGC
Accredited by NAAC with 'A' Grade, Accredited by NBA (CSE, MECH, ECE, CIVIL, BT)

A MINI PROJECT REPORT ON

APPAREL WAREHOUSE

*Submitted in partial fulfilment for the award of degree of Bachelor of
Engineering In*

COMPUTER SCIENCE AND ENGINEERING

Submitted by

POOJA SUNIL JADHAV

1NH18CS140

4B

Reviewed by

Ms. SOJA RANI

Assistant Professor, CSE

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

(Autonomous Institution Affiliated to VTU & Approved by AICTE)

Accredited by NAAC 'A', Accredited by NBA

Outer Ring Road, Panathur Post, Kadubisanahalli,

Bangalore – 560103



NEW HORIZON COLLEGE OF ENGINEERING

Autonomous College Permanently Affiliated to VTU, Approved by AICTE & UGC
Accredited by NAAC with 'A' Grade, Accredited by NBA (CSE, MECH, ECE, CIVIL, BT)

Certificate

This is to certify that the mini project titled

APPAREL WAREHOUSE

Submitted in partial fulfilment for the award of degree of Bachelor of
Engineering In

COMPUTER SCIENCE AND ENGINEERING

POOJA SUNIL JADHAV

1NH18CS140

During the academic year 2019-2020

Signature of reviewer

Signature of HOD

Semester End Examination

Name of Examiner

Signature with date

1.

.....

2.

.....

ABSTRACT

Whether you sell casual wear for fashionistas or occupational gear for workers in hazardous environments, there's no shortage of challenges facing the apparel or footwear warehouse manager. It takes the right mix of organizational skills, people skills and work ethic to manage a warehouse successfully, and the rapidly changing nature of fashion keeps even the steeliest managers on their toes. If you manage a warehouse for a fashion, footwear or sewn products business, chances are you may have stressed about one or more of these common apparel industry headaches.

- Apparel warehouse is a project that uses the concept of objects oriented programming (OOP) in java to create an inventory that assures the smooth and efficient working of the apparel warehouse.
- Apparel warehouse helps to create a list moreover an inventory that helps the warehouse to organise the apparels efficiently.
- **Encapsulation** is achieved when each object keeps its state **private**, inside a class. This will help to keep a track of the collections of the apparels that are yet to go public.
- **Abstraction** can be thought of as a natural extension of encapsulation.
- In object-oriented design, programs are often extremely large. And separate objects communicate with each other a lot. So maintaining a large codebases like this for years — with changes along the way — is difficult. Abstraction is a concept aiming to ease the above stated problem.
- Allocation is an especially important tool, as not all customer orders need to be fulfilled immediately.
- The above mentioned problems are satisfied by the working of apparel warehouse.

ACKNOWLEDGEMENT

The satisfaction and euphoria that accompany the successful completion of any task would be impossible without the mention of the people who made it possible, whose constant guidance and encouragement crowned our effort with success.

I have great pleasure in expressing my gratitude to **Dr. Mohan Manghnani**, Chairman of New Horizon Educational Institutions for providing us with the necessary infrastructure and creating good environment

I would like to thank the Principal, **Dr.Manjunatha**, New Horizon College of Engineering, for providing necessary means and a magnanimous environment.

I would like to take this opportunity to express my gratitude to **Dr.Prashanth C.S.R**, the Dean of Academics, for providing us with the various tools to implement our project.

I would like to take this opportunity to thank **Dr.B.Rajalakshmi**, Head of Department of Computer Science and Engineering for her undivided support given to all of us.

I would like to thank my reviewer Asst.Professor **Ms. Soja Rani**, my reviewer for being extremely patient with me and constantly providing me with her guidance and knowledge to go about with my mini project.

Last but not the least thank you to all the teaching and the non-teaching staff of Computer Science and Engineering Department and my friends for their cooperation and help extended to me, who helped me directly or indirectly in this course of project work.

POOJA SUNIL JADHAV [1NH18CS140]

TABLE OF CONTENTS

| SL NO | HEADINGS | PAGE NO |
|-------|----------------------------------|---------|
| | ABSTRACT | i |
| | ACKNOWLEDGEMENT | ii |
| | TABLE OF CONTENTS | iii |
| | TABLE OF FIGURES | iv |
| 1 | INTRODUCTION | 1-4 |
| | 1.1 Problem Definition | 1 |
| | 1.2 Objectives | 1 |
| | 1.3 Methodology | 2 |
| | 1.4 Expected Outcomes | 3 |
| | 1.5 Requirements Specifications | 4 |
| 2 | OBJECT ORIENTED PROGRAMMING(OOP) | 5-14 |
| | 2.1 Class | 6 |
| | 2.2 Object | 7 |
| | 2.3 Inheritance | 8-9 |
| | 2.4 Polymorphism | 10 |
| | 2.5 Abstract Class | 11 |
| | 2.6 Multi-Threading | 12 |
| | 2.7 I/O Functions | 13 |
| | 2.8 Exception | 14 |
| 3 | DESIGN | 15-17 |
| | 3.1 Goal | 15 |
| | 3.2 Algorithm | 16-17 |
| 4 | Implementation | 18-23 |
| | 4.1 Module 1 | 18 |
| | 4.2 Module 2 | 19 |
| | 4.3 Module 3 | 20 |
| | 4.4 Module 4 | 21 |
| | 4.4 Built-in Functions | 21 |
| | 4.6 Switch case Statement | 22 |
| | 4.7 Loops | 23 |
| 5 | RESULT | 24-35 |
| 6 | CONCLUSION | 36 |
| 7 | BIBLIOGRAPHY | 37 |

TABLE OF FIGURES

| SL NO | TOPIC | PAGE NO |
|-------|-----------------------------|---------|
| 1 | Object-oriented programming | Page 5 |
| 2 | Class | Page 6 |
| 3 | Object | Page 7 |
| 4 | Inheritance | Page 9 |
| 5 | Polymorphism | Page 10 |
| 6 | Multi-threading | Page 12 |
| 7 | I/O Function | Page 13 |
| 8 | Exception | Page 14 |
| 3 | FIG 5.1 | Page 24 |
| 4 | FIG 5.2 | Page 25 |
| 5 | FIG 5.3 | Page 26 |
| 6 | FIG 5.4 | Page 27 |
| 7 | FIG 5.5 | Page 28 |
| 8 | FIG 5.6 | Page 29 |
| 9 | FIG 5.7 | Page 30 |
| 10 | FIG 5.8 | Page 31 |
| 11 | FIG 5.9 | Page 32 |
| 12 | FIG 5.10 | Page 23 |
| 13 | FIG 5.11 | Page 34 |
| 14 | FIG 5.12 | Page 35 |

CHAPTER 01

INTRODUCTION

1.1 PROBLEM STATEMENT

Whether you sell casual wear for fashionistas or occupational gear for workers in hazardous environments, there's no shortage of challenges facing the apparel or footwear warehouse manager. It takes the right mix of organizational skills, people skills and work ethic to manage a warehouse successfully, and the rapidly changing nature of fashion keeps even the steeliest managers on their toes. If you manage a warehouse for a fashion, footwear or sewn products business, chances are you may have stressed about one or more of these common apparel industry headaches.

1.2 OBJECTIVES

- Apparel warehouse is a project that uses the concept of objects oriented programming (OOP) in java to create an inventory that assures the smooth and efficient working of the apparel warehouse.
- Apparel warehouse helps to create a list moreover an inventory that helps the warehouse to organise the apparels efficiently.
- **Encapsulation** is achieved when each object keeps its state **private**, inside a class. This will help to keep a track of the collections of the apparels that are yet to go public.
- **Abstraction** can be thought of as a natural extension of encapsulation.
- In object-oriented design, programs are often extremely large. And separate objects communicate with each other a lot. So maintaining a large codebases like this for years — with changes along the way — is difficult. Abstraction is a concept aiming to ease the above stated problem.
- Allocation is an especially important tool, as not all customer orders need to be fulfilled immediately.
- The above mentioned problems are satisfied by the working of apparel warehouse.

1.3 METHODOLOGY

Apparel warehouse uses the concept of Object Oriented Programming (OOP) to create a medium that helps make an inventory of the interested piece of garment that is to be purchased. Cost accounting information is designed for managers. Since managers are making decisions only for their own organization, there is no need for the information to be comparable to similar information from other organizations. Instead, the important criterion is that the information must be relevant for decisions that managers, operating in a particular environment of business including strategy, make. Cost accounting information is commonly used in financial accounting information, but first we are concentrating on its use by managers to take decisions. The accountants who handle the cost accounting information add value by providing good information to managers who are making decisions. Among the better decisions, is the better performance of one's organization, regardless if it is a manufacturing company, a bank, a non-profit organization, a government agency, a school club or even a business school? The cost-accounting system is the result of decisions made by managers of an organization and the environment in which they make them.

1.4 EXPECTED OUTCOME

Stores doing Omni channel retailing are at the top of their game; they attract the 90% of consumers who switch between at least three applications per day to complete specific tasks.

And that's minus consumers who shop physically.

Imagine profits gushing in from several online channels and from your physical store too. But that profit advantage comes with a serious inventory control responsibility — Omni channel inventory management.

To phase this responsibility, brands now include solid inventory control plans for sales coming through multiple selling channels.

In fact, Joel Beal, CEO and co-founder of Alloy.ai, predicts that in 2019, brands will capitalize more on sophisticated data analysis methods to leverage store-level data. He says:

“Companies are becoming increasingly attuned to the importance of using store-level data to customize supply to match local demand. With sophisticated data analysis options now available, brands and retailers will capitalize on the opportunity to improve sales.”

Armed with such concrete and advanced-level data, you can control your store supply chains, distribution channels, and avoid stock outs this year.

However, as you start attracting sales from multiple channels, it becomes a tad more complex to control inventory — given that you'll be serving way more customers than you used to (depending on how much Omni channel grows your business).

2.2 REQUIREMENTS AND SPECIFICATIONS

- SOFTWARE REQUIREMENTS:
 1. **Compilers:** Blue Eclipse, or more.
 2. **Operating System: Windows** 10 or more.
- HARDWARE REQUIREMENTS:
 1. Processor: Intel Pentium 4 or more.
 2. Ram: 1 GB or more.
 3. Hard Disk: 120 GB hard disk.

CHAPTER 02

OBJECT ORIENTED PROGRAMMING (OOP)

Object Oriented Programming (OOP) is a type of programming language that is object oriented. It uses a bottom up approach. There are various features of OOP that make it more favourable to work with and create a higher edge over POP.

The various features of OOP are:

- Inheritance.
 - Encapsulation,
 - Polymorphism.
 - Abstraction. Etc...
- These are only the few to name.

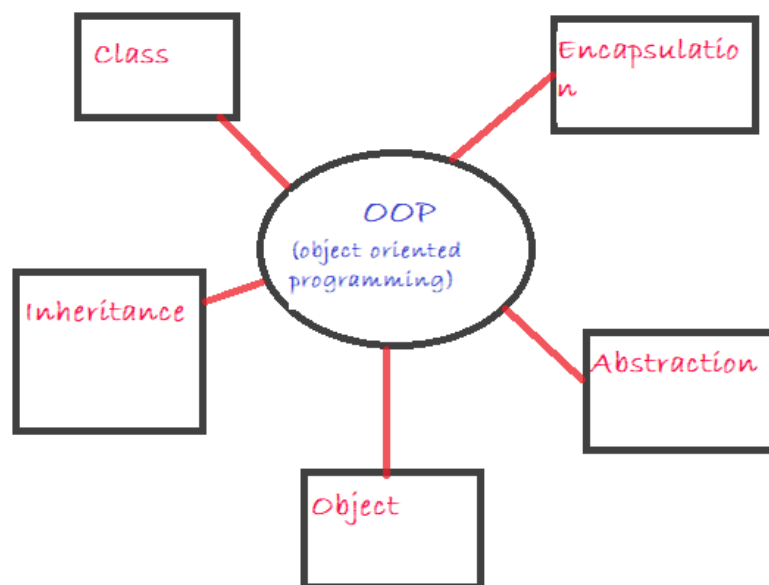


Fig 1: oop

2.1 Class:

A class is a user defined blueprint or prototype from which objects are created. It represents the set of properties or methods that are common to all objects of one type. In general, class declarations may or may not have the following:

1. **Modifiers:** A class can be public or has default access.
2. **Class name:** The name should begin with an initial letter.
3. **Superclass:** The name of the class's parent (superclass or base class), if any, preceded by the keyword extends. A class can only extend (subclass or base class) one parent.
4. **Interfaces:** A comma-separated list of interfaces implemented by the class, if any, preceded by the keyword implements. A class can implement more than one interface.
5. **Body:** The class body surrounded by braces, { }.

The following image gives us a pictorial representation of a class.

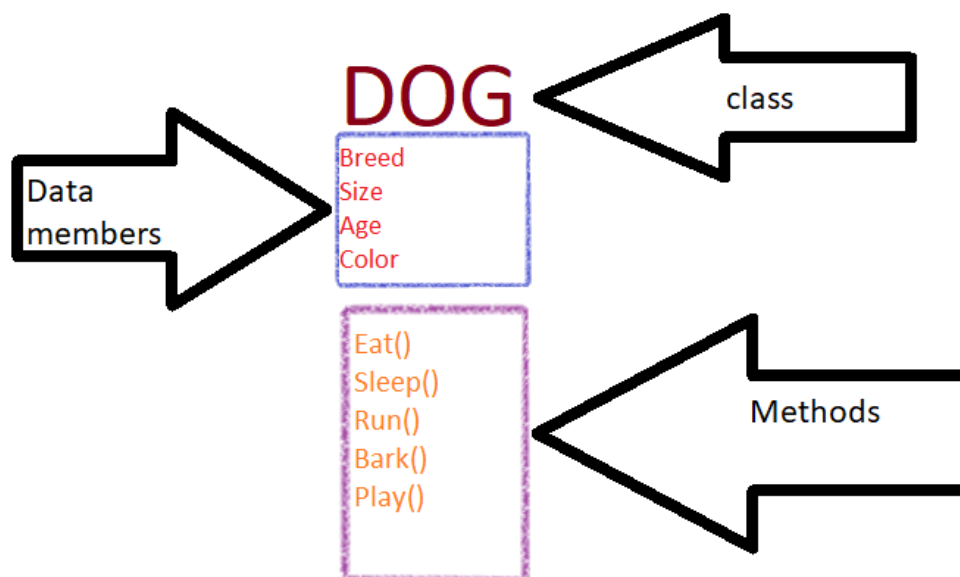


Fig 2: Class in java

2.1 Object:

It is a basic unit of Object Oriented Programming and represents the real life entities. A typical Java program creates many objects, which as you know, interact by invoking methods. An object consists of:

1. **State:** It is represented by attributes of an object. It also reflects the properties of an object.
2. **Behaviour:** It is represented by methods of an object. It also reflects the response of an object with other objects.
3. **Identity:** It gives a unique name to an object and enables one object to interact with other objects.
4. Objects correspond to things found in the real world. For example, a graphics program may have objects such as “circle”, “square”, “menu”. An online shopping system might have objects such as “shopping cart”, “customer”, and “product”.

Declaring Objects (Also called instantiating a class):

When an object of a class is created, the class is said to be **instantiated**. All the instances share the attributes and the behaviour of the class. But the values of those attributes, i.e. the state are unique for each object. A single class may have any number of instances.

As we declare variables like (type name ;). This notifies the compiler that we will use name to refer to data whose type is type. With a primitive variable, this declaration also reserves the proper amount of memory for the variable. So for reference variable, type must be strictly a concrete class name. In general, we **can't** create objects of an abstract class or an interface.

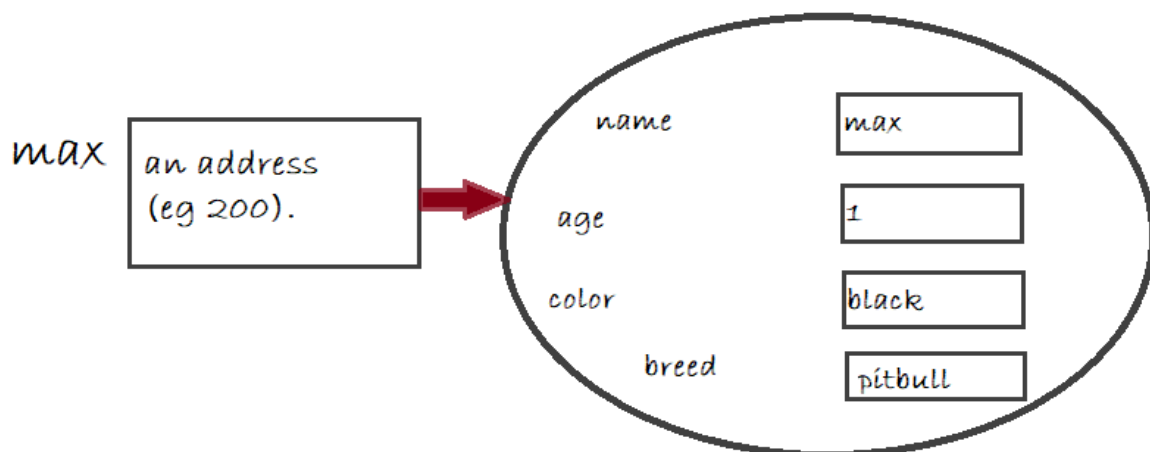


Fig 3: Object.

2.3 Inheritance:

Inheritance is an important pillar of OOP (Object Oriented Programming). It is the mechanism in java by which one class is allows inheriting the features (fields and method of another.

- **Super Class:** The class whose features are inherited is known as super class (or a base class or a parent class).
- **Sub Class:** The class that inherits the other class is known as sub class (or a derived class, extended class, or child class). The subclass can add its own fields and methods in addition to the superclass fields and methods.
- **Reusability:** Inheritance supports the concept of “reusability”, i.e. when we want to create a new class and there is already a class that includes some of the code that we want, we can derive our new class from the existing class. By doing this, we are reusing the fields and methods of the existing class.

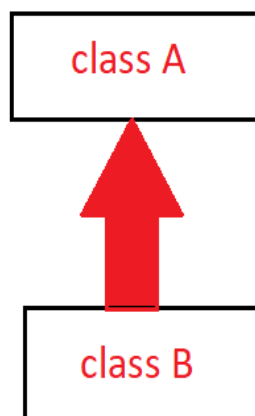


Fig 4(a): Inheritance

Types of Inheritance in Java

Below are the different types of inheritance which is supported by Java.

Single Inheritance: In single inheritance, subclasses inherit the features of one superclass. In image below, the class A serves as a base class for the derived class

Multilevel Inheritance: In Multilevel Inheritance, a derived class will be inheriting a base class and as well as the derived class also act as the base class to other class. In below image, the class a serves as a base class for the derived class B, which in turn serves as a base class for the derived class C. In Java, a class cannot directly access the grandparent's members.

Hierarchical Inheritance: In Hierarchical Inheritance, one class serves as a superclass (base class) for more than one sub class. In below image, the class A serves as a base class for the derived class B, C and D.

Multiple Inheritances (Through Interfaces): In Multiple inheritances, one class can have more than one superclass and inherit features from all parent classes. Please note that Java does **not** support multiple inheritances with classes. In java, we can achieve multiple inheritances only through Interfaces. In image below, Class C is derived from interface A and B.

Hybrid Inheritance (Through Interfaces): It is a mix of two or more of the above types of inheritance. Since java doesn't support multiple inheritances with classes, the hybrid inheritance is also not possible with classes. In java, we can achieve hybrid inheritance only through Interfaces.

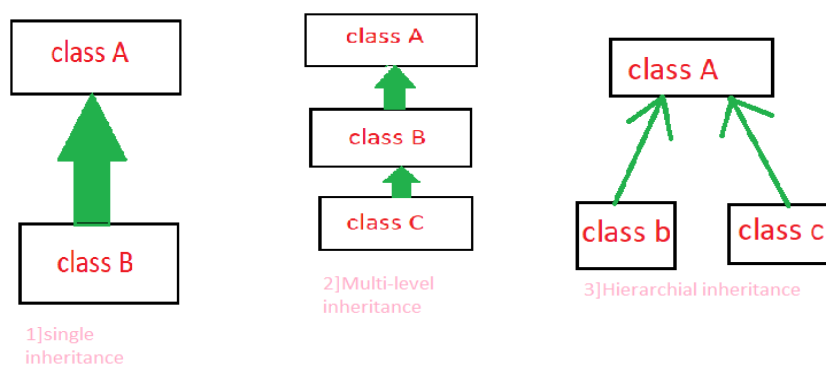


Fig 4(b): Various Inheritances.

2.4 Polymorphism:

Polymorphism is a feature that allows one interface to be used for a general class of actions.

The specific action is determined by the exact nature of the situation.

It allows two or more classes to respond or behave to the same message/command in various different ways.

Basically it means that the user can use the same name for a method in two or various ways in the same class or different classes. It also means that the user can send the same message to two different classes and still get the correct response. Polymorphism in java is usually obtained by **Function or Method Overloading**.

In Method overloading the Method or the Function used that helps to define how the class data can be used, so they will usually be two methods with the same name but different parameters or to be more precise signatures.

Below is a simple example of method overloading.

Colour ("ombre","aqua");

Colour ("ombre","aqua","beige");

Here there are two methods with the same name colour but the only difference that is clearly seen is that the above or the first colour take two parameters which is "ombre" and "aqua"

Whereas

the second colour takes three parameters which are "ombre", "aqua" and "beige".

The point of Method Overloading is that it allows the flexibility to call two different methods.



Fig 6: Polymorphism

2.5 Abstract Class:

An essential element of OOP is abstraction. Human's manage complexity through abstraction.

For example, people do not think of a car as set of tens of thousands of individual parts. They think of it as a well-defined object with its unique behaviour. This abstraction allows people to drive to the department stores without being overwhelmed by the complexity of the parts of the car.

Abstraction is to represent essential of a system without getting involved with the complexity of the entire system.

Abstraction in java is achieved by creating abstract class in the program.

Whenever a class is declared abstract the class cannot instantiate .it means that the class cannot have any objects .it can have methods and these methods have to be implemented or moreover instantiated in the sub class if they exist.

The main use of abstraction is that the user can create an interface without actually instantiating it. Abstraction helps to create a synopsis of the program what it actually does without diving deep into the details of the program. This allows the user to receive a surface level knowledge of the program.

Abstraction can be viewed or better understood by taking the example of the synopsis submission of our mini project to our reviewer.

Here it gives a glimpse or surface level knowledge to the reviewer about the mini project without diving deep into the working of the mini project.

A class is made abstract by using the keyword "abstract".

Example:

```
abstract class mini
```

```
{
```

```
    _____
```

```
    _____ //body of the class that is not instantiated.
```

```
    _____
```

```
}
```

2.6 Multi-Threading:

Mutely-threading is defined as the simultaneous execution of two or parts of a program. Multithreading allows the maximum utilization of the central processing unit CPU.

Every part of the above mentioned program is called a thread.

So multi-threading is technically a light weight process inside another one.

There usually two ways of creating a multithreading:

- 1: Extending the thread class.
2. Implementing the runnable class.

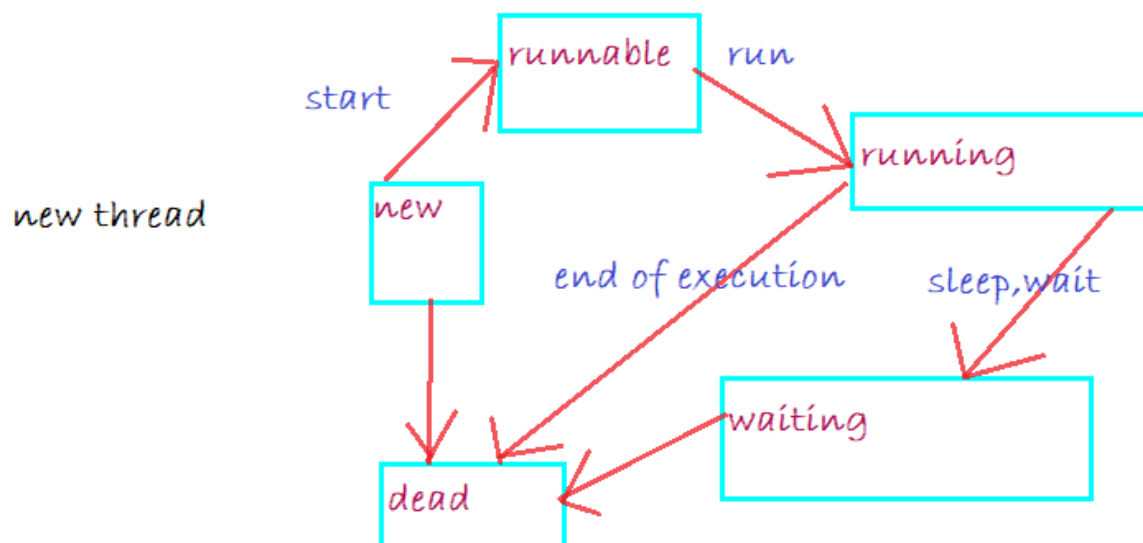


Fig 7:Multi-Threading in Java

2.7 I/O Functions:

Scanner Classes:

Scanner class is a class in java.util, which allows the user to read (input) values of the various types.

The syntax of the Scanner class is as below:

```
Scanner sn =new Scanner(System.in); //System.in is an InputStream.
```

The Scanner looks for tokens in the input. A token is a series of characters that ends with what Java calls “whitespace”. A whitespace character can be blank, a tab character, a carriage return, or the end of the file.

Printer class:

The java.awt.print package contains classes and interfaces that support printing. The most important class in this package is PrintJob; it coordinates the printing process. The Printable and Pageable interfaces represent printable objects or documents that can be printed with a PrinterJob.

Java.awt.print class implements a multi-page Pageable document out of individual single page Printable objects. Each page of the Book can have its own Page format.

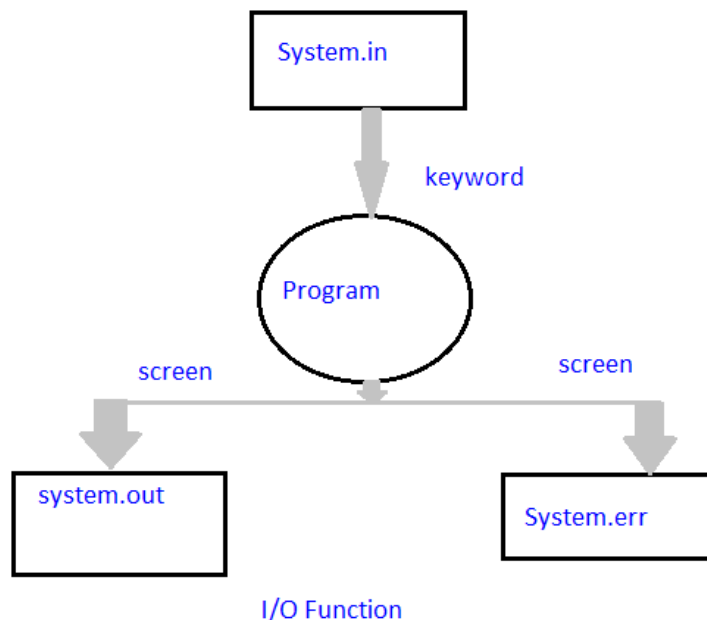


Fig 8.

2.8 Exception:

an exception is a run time error in a Java program that is caused because some abnormal condition getting executed in the code. Exception in java can be either manually created or they can exists.

One should not get confused with an error and an exception. These are different things in java.

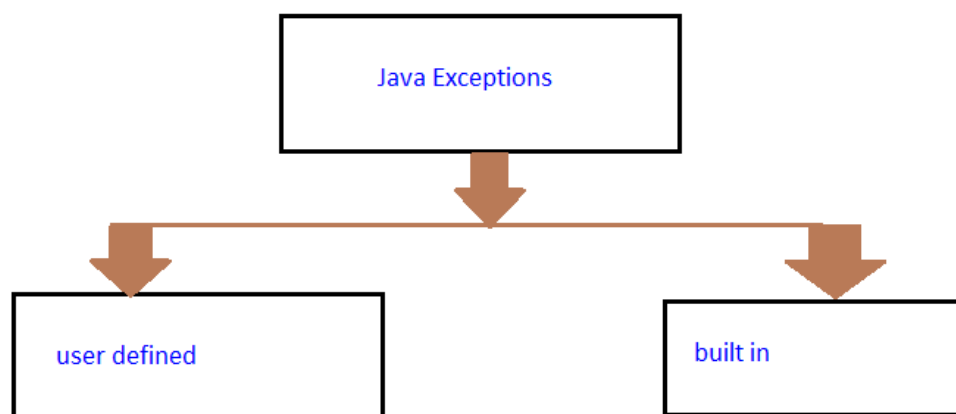
Error: it's a problem(serious one) that is caused and no reasonable application should try to catch it.

Exception: it's a serious problem caused that might or can be caught by a reasonable application.

All exceptions and errors are the sub class of "Throw able" class.

There are various keywords that are used to create an exception. The few or one can say the most prominent of them are:

1. catch
2. throws
3. throw
4. final
5. finals. Etc..



EXCEPTIONS IN JAVA

Fig 9.

CHAPTER 03

DESIGN

3.1 Goal:

The main aim of Apparel Warehouse is to provide a more efficient, easy and compact way of cataloguing. This project uses the various object oriented programming principles that we've learned during the course of this semester and includes the necessary ones that aid the project.

Not only does this act as a catalogue but it also is means of buying the products that the user likes when they see the catalogue and can buy them.

apparel warehouse provides a selection of clothes and accessories according to different collections based on the seasons of spring ,summer winter.it gives the user an option to see and also by custom made apparels or to be more precise"couture".the warehouse also gives an option of daily wear that the user can see or buy if they want to.

The project uses only specified principles of object oriented programing language.

Apparel warehouse eases the lengthy process of differentiating or categorizing the apparels in a more sufficient manner.

The whole aim would be if I were to summarise it is that apparel warehouse provides a better ,faster, efficient flexible catalogue and a shop zone.

3.1 Algorithm/Pseudo code:

- Step 1: START (begin)

- Step 2:

This display the option whether to do the following:

- A. Press 1-6 for summer collection.
- B. Press 7-12 for spring collection.
- C. Press 13 for pre-fall collection.
- D. If the options A or B are selected then the program asks for couture or daily.
- E. Accordingly in takes the user to the daily or couture section.
- F. Displays another menu asking the user if they would wish to buy anything or would just want to exit.
- G. Exit.

- Step 3: the following events take place:

The predefined menu is displayed with 3 options of various collections at fix and standard prices.

The user is provided with an option of 3 cases:

CASE 1: Summer collection:

In the above case the user is asked whether they would wish to select from the couture or daily collection. Respective of their choices the specified collection is displayed.

CASE 2: Spring collection:

This also provides the same options like the summer collection does like whether they would wish to select from the couture or daily collection. Respective of their choices the specified collection is displayed. The main difference is that the add-ons are available and the catalogue suits the spring needs.

CASE 3: Pre-fall collection:

This collection unlike the spring and the summer collection do not provide a daily option; it only provides a couture collection owing to its type.

- Step 4:

A prompt option appears and asks the user whether they want to Continue the above steps by pressing 'y' or 'Y' to continue.

If yes then the above steps are repeated for 5 times only.

The control of the program goes back again to the start and the statement "Welcome to house of Gianni" is displayed again.

The whole process from the beginning is executed all over again.

- Step 5:

In the above step the user is asked whether they wish to buy or just leave the "Gianni house". If 'y' or 'Y' is pressed then the menu for the collection number including the couture and the daily collections are displayed and according the user can place their orders. so the invoice or the details of the collection along with its respective prices are displayed. The user can place only three orders. After doing the above the bill for the whole expenditure is generated.

Then finally the "DO YOU WANT TO CONTINUE? PRESS YES OR NO" prompt is displayed again. If no is the answer then step 6 is executed

Else step 5 is executed which is the exit option.

- Step 6: STOP (exit)

This function brings the user out of the loop.

This happens if the user presses any key apart from the "Y" or "y" key.

You can press literally any key because the condition of the loop will be true until "Y" or "y" is pressed.

This happens due the exit control feature of the do while loop.

Due to this functionality the loop is at least executed once even when the condition becomes false.

The reason being that the condition is checked at the end.

CHAPTER 04

IMPLEMENTATION

4.1 Module 1

A) Menu:

This module provides the user with options ranging from 1-13, where 1-6 is used to display the summer collection, 7-12 is used to display the spring collection and 13 is used to display the pre-fall collection.

Here the user is asked their choice and according the message is displayed.

For the spring and summer collections the daily or couture options are displayed.

For pre-fall only couture options are available and the note is displayed that says the same.

Here if the options apart from 1-12 the remains keys are considered invalid and a message is displayed prompting the user to enter the right options.

b) Sub-menu:

This sub menu part allows the user to select whether they wish to choose and see the couture or the daily catalogue.

So these sub menus have the specified type of the specified season collection .after this once the decision is made then the catalogue is displayed with various fields that gives the user the information about the collections and the contents that come with it.

The various fields are:

- 1) Material: denotes the type of fabric is of and whether it's eco-friendly.
- 2) Authenticity: the originality check for the clothes.
- 3) Season: specifies the season.
- 4) Style.
- 5) Add-ons: this specifies the various accessories available with the collection.
- 6) Price. And so on...

4.2 Module 2: Classes.

This module mainly focuses on the classes that are present in the apparel warehouse project:

The first class is Versace.

This class acts as the parent or base/super class from where through the help of extend keyword creates inheritance of single type to another class. This class helps the sub class to have methods and various variables that Versace has and aids in the reusability of the code, makes the program compact and precise.

The class has two constructors that help in creating objects of the class. The class mainly focuses on creating the daily catalogue for both the summer and spring collections. The constructors experience overloading hence emerging as the polymorphism aspect of the principles of object oriented programming language.

The constructors are differentiated by the addition of the add-on parameter in the signature of the constructor that creates an object for spring daily collection.

It has a method called inventory that creates moreover helps to create a detailed print for the collection that has been chosen.

The next class is Gianni.

The above class is called the sub/child/derived class for the above class mentioned above. The class focuses on creating objects for the couture category for all the three collections .the class has three constructors again aiding in polymorphism. The constructors have different signatures.

This class has methods to display the detailed account of the collection and also has the menu driven part of the program.

All the loops and the switch case statements are present in this class. Eventhough this class is the derived class but the major chunk of the code lies in this class. The class has a well-defined introduction page for the program.

4.3 Module 3: Constructors:

It can be tedious to initialize all of the variables in a class each time an instance created. It would be simpler and more concise to have all of the setup done at the object is first created.

Java allows objects to initialize themselves when they are created. This automatic initialisation is performed through the use of a constructor.

When we create an object of a class, a special kind of method called constructors is always invoked .a constructor initializes an object immediately upon creation.

There are a total of 5 constructors that are used in this project.2 constructors are of class Versace and the reaming three are of class Gianni.

The primary purpose of constructors is initializing the instance variable uniquely for the object that is being created. A constructor has two special characteristics which differentiate it from other class methods.

- 1) A constructor never returns a value and the programmer must not specify a return type- not even of type void.
- 2) A constructor always has the same name as the class.
- 3) A constructor is automatically called immediately after the object is created, before new operator completes.

There are mainly two types of constructors:

- 1) Default constructors.
- 2) parameterised constructor (Constructors with arguments).

Default constructors:

If there is no constructor defined by the programmer, then the compiler crates a default constructor for the class. The default constructors automatically initialize all instances variables to zero, Boolean to false and string to null. Once a constructor is defined for the class the default constructor is no longer used.

Parameterised constructor (Constructors with arguments):

It may be required to initialize the various data members of different with different values .in that case, constructors can be created with parameters. Parameters will initialize the instance variables based on the argument passed to it constructors that has parameters is called as the above.

4.4 Module 4: various objects used.

There are a total of the five objects that have been created in this project.

So the summer and the spring collections have two objects each for daily and couture respectively. The pre fall collection has only one object because for the reason that the collection doesn't contain the daily option.

The object is created using the constructors.

Theses act as attributes to the two classes.

4.4 Built in functions.

The built function in this project is the inventory function and bill function.

The inventory acts like a print of the details of the collections that are viewed by the user.

It is a method that returns a type void .method overloading is achieved by creating the inventory methods with various different signatures hence polymorphism is expressed through it.

A simple example of the above would be displayed in the sample output section.

Then we have the method that simply displays the bill of the total of the purchases made.

4.6 Switch Case Statement

We are using switch case statements in the program to perform the operations as per the choice of the user.

The user is asked to enter the choice number which is passed in the switch statement.

Based on the value, the functions are called within the case statements.

When a case matches with the option entered, the function is called followed by the break statement.

The break statement is used to come out of the switch statement.

When the break statement is not given at the end of each case, then all the cases are executed.

At the end of the case statements, a default statement has been included to give an error message when the user enters the value which is not in the option list.

The syntax of the switch statement is:

```
switch (n)
{
    case 1: //execute the following if the option matches the case.
        Break;

    case 2: //execute the following if the option matches the case.
        break;

    default: // execute the following if none of the cases matches the option entered.
        break;
}
```

4.7 Loops:

i) While loop

The while loop is used to control the flow of execution of the program. The while loop is being used in the program to figure out the exact position where the data must be entered.

While loop used in insertion function:

The while loop that is being used in the insertion function checks if the node points to NULL and if the priority of the node that the current node points to is higher than the priority that the user has entered.

The compiler comes out of the while loop only when either of the conditions is false, otherwise the pointer is made to point to the next node in the linked list.

The while loop used in display:

The while loop used in the display function is executed until the pointer is pointing to NULL.

ii) The do-while loop

The do while loop is used when we want to execute a set of code at least once. Once a set of code is executed, the condition is checked.

The do while loop in this function displays the option available to the student at least once. When the student is done entering the details or checking the details, then the condition is checked at the end ,that is, if the student wants to continue then they enter yes(y), in which case the loop continues to execute the same set of statements.

If the user enters no (n), then we come out of the loop and the rest of the program is executed.

The do-while loop in the menu function-The functioning of the do while loop in this function is like its functioning in the display function.

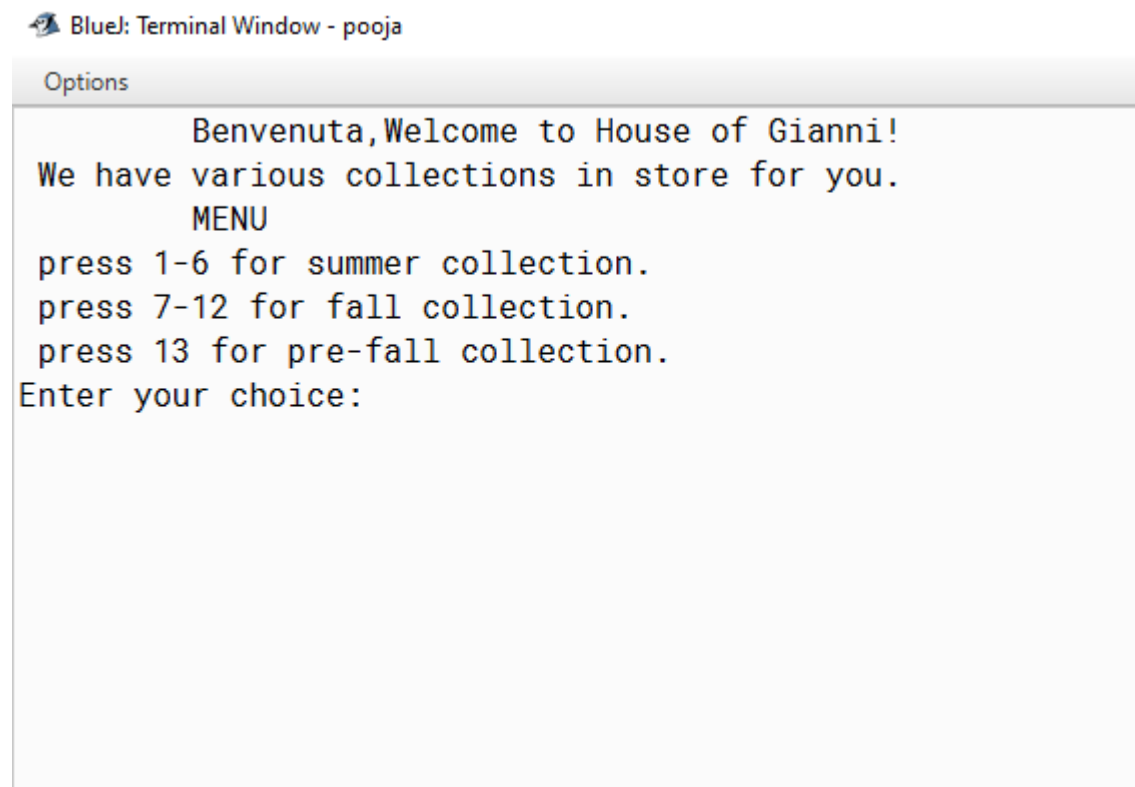
CHAPTER 05

SAMPLE OF THE OUTPUT

This section of the report contains the various outputs that are possible in the program.

Let's assume that a user name Wanda is using the warehouse and following are her activities.

1. The beginning of the program.



```
BlueJ: Terminal Window - pooja
Options
Benvenuta,Welcome to House of Gianni!
We have various collections in store for you.
MENU
press 1-6 for summer collection.
press 7-12 for fall collection.
press 13 for pre-fall collection.
Enter your choice:
```

FIG 5.1

The below figure is represents when the output when summer collection of type daily is chose.

```
.
press 13 for pre-fall collection.
Enter your choice:
2
    Welcome to the summer collection.
would you like to choose couture or daily
daily
Welcome to the daily wear
This is the list of items available under dailywear
    romper by bella
    crop top by adonis
    white sneakers by hans
The inventory list for the piece is as follows:
SEASON: season
STYLE: daily
TYPE: romper @99.00
SHADE: pastel pink
SHOES: white sneakers @45.6
MATERIAL: cotton
AUTHENTICITY: romper by BELLA;shoes by HANS;sunglasses by MAX
REFERENCE NUMBER: dsu2019
PRICE: 209.6
Do you want to continue?/n Press [y/Y]es or [n/N]o /n Please enter your choice:/n
```

FIG 5.2

The below figure is represents when the spring option that is chose. With the remaining options that follows it.

The predefined menu.

```
Do you want to continue?/n Press [y/Y]es or [n/N]o /n Please enter your choice:/n
y
    MENU
    press 1-6 for summer collection.
    press 7-12 for fall collection.
    press 13 for pre-fall collection.
Enter your choice:
7
    Welcome to the spring collection.
    would you like to choose coture or daily
couture
    The inventory list for the piece is as follows:
    SEASON: spring
    STYLE: couture
    ADDON: trenchcoat
    TYPE: tuxedo+ trenchcoat(long and carmel brown)
    SHADE: ombre
    MATERIAL: leather
    AUTHENTICITY: faux pas by MUIGLARE
    REFERENCE NUMBER: csp2019
    PRICE: 465.55
Do you want to continue?/n Press [y/Y]es or [n/N]o /n Please enter your choice:/n
```

FIG 5.3

The below picture denotes the second choice that the user wishes to see.

```
REFERENCE NUMBER: cpre2020
PRICE: 465.55
Do you want to continue?/n Press [y/Y]es or [n/N]o /n Please enter your choice:/n
y
MENU
press 1-6 for summer collection.
press 7-12 for fall collection.
press 13 for pre-fall collection.
Enter your choice:
13
pre-fall
NOTE:only couture is available
The inventory list for the piece is as follows:
SEASON: prefall
STYLE: couture
ADD-ON: trenchcoat
CARRY-ON: umbrella
TYPE: overalls+black tee+umbrella(dark grey)
SHADE: sooty grey
MATERIAL: wool
AUTHENTICITY: yarn by MAEVE
REFERENCE NUMBER: cpre2020
PRICE: 565.87
Do you want to continue?/n Press [y/Y]es or [n/N]o /n Please enter your choice:/n
```

FIG 5.4

The below figure explains or tells the user to type in the right option in order for the process to move on.

```
CARRY-ON: umbrella
TYPE: overalls+black tee+umbrella(dark grey)
SHADE: sooty grey
MATERIAL: wool
AUTHENTICITY: yarn by MAEVE
REFERENCE NUMBER: cpre2020
PRICE: 565.87
Do you want to continue?/n Press [y/Y]es or [n/N]o /n Please enter your choice:/n
y
MENU
press 1-6 for summer collection.
press 7-12 for fall collection.
press 13 for pre-fall collection.
Enter your choice:
15
please enter the above mentioned options
Do you want to continue?/n Press [y/Y]es or [n/N]o /n Please enter your choice:/n
```

FIG 5.5

The below figure asks the user whether they want to view or move towards the purchase.

```
Enter your choice:
15
please enter the above mentioned options
Do you want to continue?/n Press [y/Y]es or [n/N]o /n Please enter your choice:/n
n
You have reached the end of the inventory.
Would you like to make your purchase? inorder to make the purchase please check the below menu:
NOTE: only 3 purchases can be done
press the following for the purchases
1:Couture-Summer
2:Daily:Summer
3:Couture-Spring
4:Daily-Spring
5:Pre-Fall(Couture only)
enter your options
```

FIG 5.6

The below begins with the purchase (1).

```
NOTE: only 3 purchases can be done
press the following for the purchases
1:Couture-Summer
2:Daily:Summer
3:Couture-Spring
4:Daily-Spring
5:Pre-Fall(Couture only)
enter your options

1
    The inventory list for the piece is as follows:
    SEASON: summer
    STYLE: couture
    TYPE: tuxedo
    SHADE: teal
    MATERIAL: cotton
    AUTHENTICITY: faux pas by OTIS
    REFERENCE NUMBER: csu2019
    PRICE: 532.55
Do you want to continue?/n Press [y/Y]es or [n/N]o /n Please enter your choice:/n
```

FIG 5.7

The below figure depicts second purchase.

```
press the following for the purchases
1:Couture-Summer
2:Daily:Summer
3:Couture-Spring
4:Daily-Spring
5:Pre-Fall(Couture only)
enter your options

5
    The inventory list for the piece is as follows:
    SEASON: prefall
    STYLE: couture
    ADD-ON: trenchcoat
    CARRY-ON: umbrella
    TYPE: overalls+black tee+umbrella(dark grey)
    SHADE: sooty grey
    MATERIAL: wool
    AUTHENTICITY: yarn by MAEVE
    REFERENCE NUMBER: cpre2020
    PRICE: 565.87
Do you want to continue?/n Press [y/Y]es or [n/N]o /n Please enter your choice:/n
```

FIG 5.8

The below figure denotes the third purchase.

```
Do you want to continue?/n Press [y/Y]es or [n/N]o /n Please enter your choice:/n
y
press the following for the purchases
1:Couture-Summer
2:Daily:Summer
3:Couture-Spring
4:Daily-Spring
5:Pre-Fall(Couture only)
enter your options

2
    The inventory list for the piece is as follows:
    SEASON: season
    STYLE: daily
    TYPE: romper @99.00
    SHADE: pastel pink
    SHOES: white sneakers @45.6
    MATERIAL: cotton
    AUTHENTICITY: romper by BELLA;shoes by HANS;sunglasses by MAX
    REFERENCE NUMBER: dsu2019
    PRICE: 209.6
```

FIG 5.9

Wanda decides to order all available on the menu by entering 3 as the total count of items.

After placing the order she is asked whether she wants to continue or not to which she replies yes (by pressing 'y').

The total amount of her bill is displayed.

```
enter your options
3
    The inventory list for the piece is as follows:
    SEASON: spring
    STYLE: couture
    ADDON: trenchcoat
    TYPE: tuxedo+ trenchcoat(long and carmel brown)
    SHADE: ombre
    MATERIAL: leather
    AUTHENTICITY: faux pas by MUIGLARE
    REFERENCE NUMBER: csp2019
    PRICE: 465.55
Do you want to continue?/n Press [y/Y]es or [n/N]o /n Please enter your choice:/n
n
your final bill is :1168.11
thank you for shopping at Gianni!
```

FIG 5.10

This figure represents the output when Wanda wants to just look through the warehouse and not buy anything.

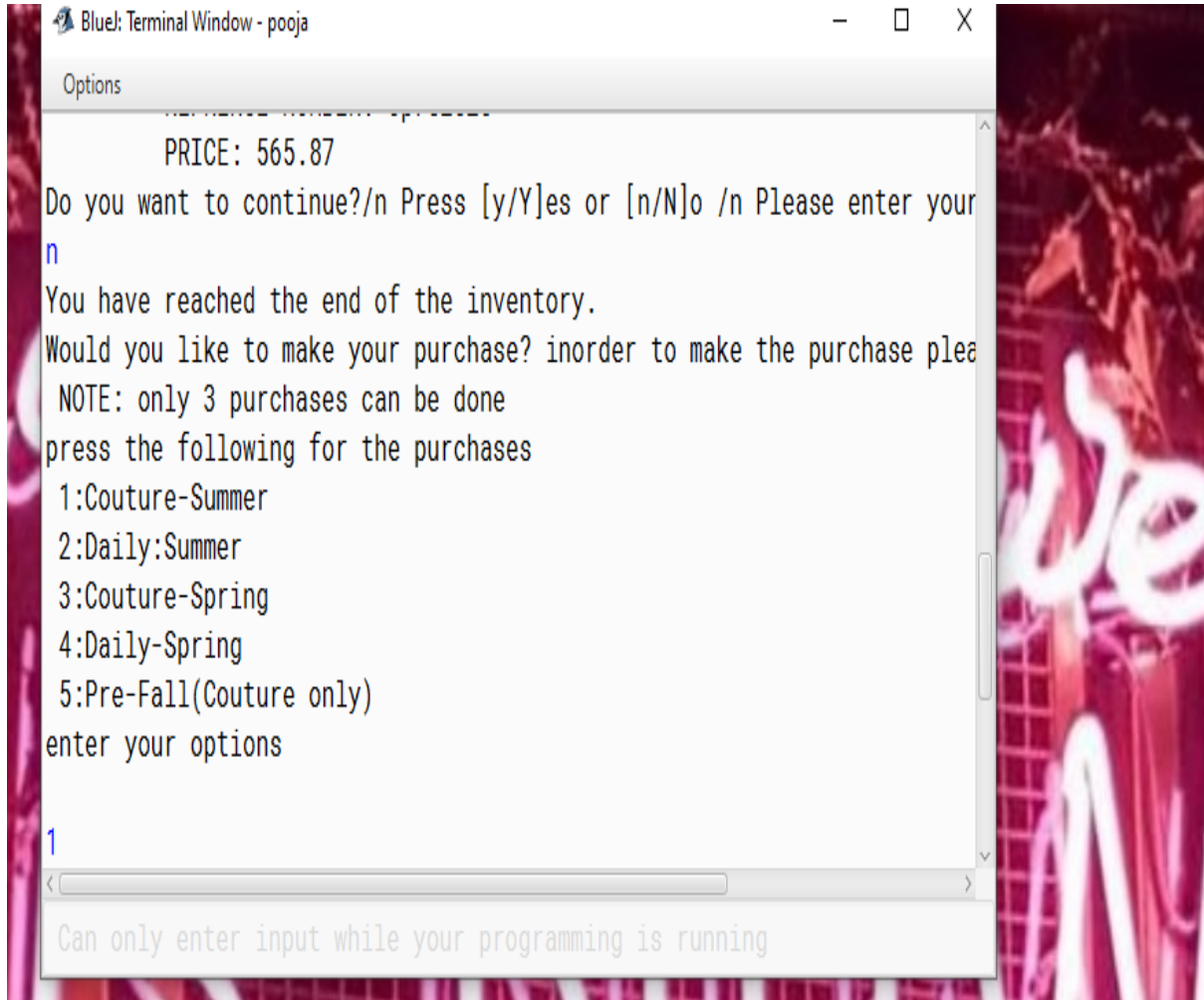
```
press the following for the purchases
1:Couture-Summer
2:Daily:Summer
3:Couture-Spring
4:Daily-Spring
5:Pre-Fall(Couture only)
enter your options

1
    The inventory list for the piece is as follows:
    SEASON: summer
    STYLE: couture
    TYPE: tuxedo
    SHADE: teal
    MATERIAL: cotton
    AUTHENTICITY: faux pas by OTIS
    REFERENCE NUMBER: csu2019
    PRICE: 532.55
Do you want to continue?/n Press [y/Y]es or [n/N]o /n Please enter your choice:/n
n
thank you for shopping at Gianni!

Can only enter input while your programming is running
```

FIG 5.11

When Wanda presses the no ('n') option the program ends there and displays this



```
Blue! Terminal Window - pooja
Options
PRICE: 565.87
Do you want to continue?/n Press [y/Y]es or [n/N]o /n Please enter your
n
You have reached the end of the inventory.
Would you like to make your purchase? inorder to make the purchase plea
NOTE: only 3 purchases can be done
press the following for the purchases
1:Couture-Summer
2:Daily:Summer
3:Couture-Spring
4:Daily-Spring
5:Pre-Fall(Couture only)
enter your options
1
Can only enter input while your programming is running
```

FIG 5.12

CHAPTER 05

CONCLUSION

The sole purpose of the project is to provide a better way to view catalogues of the various collection offered by the warehouse which are at one's disposal provided that they can splurge the cash on these goods.

The project provides details, very precise ones about the apparels available.

It aims at making the so called "just looking through" a faster process.

It also makes the warehouse to be more compact and so provide a detailed note about their various in house collections.

CHAPTER 06

BIBLIOGRAPHY

The figures used to represent the various object oriented programming principles are done in paint but using Google pic as the reference.

The written content belongs to the data oops in java class notes.

The content is formed by studying the topics with the help of geeks for geeks website.

Major webpages visited is mentioned below.

//<http://www.google.com>

Apparel Warehouse

ORIGINALITY REPORT

6%

SIMILARITY INDEX

%

INTERNET SOURCES

6%

PUBLICATIONS

%

STUDENT PAPERS

PRIMARY SOURCES

1

Kishori Sharan. "Beginning Java 9 Fundamentals", Springer Science and Business Media LLC, 2017

Publication

1%

2

A. Prasad Sistla, Ouri Wolfson, Yelena Yesha, Robert Sloan. "Towards a theory of cost management for digital libraries and electronic commerce", ACM Transactions on Database Systems, 1998

Publication

1%

3

Tan Kiat Shi, Willi-Hans Steeb, Yorick Hardy. "SymbolicC++", Springer Science and Business Media LLC, 2000

Publication

1%

4

Akshay Bharadwaj K H, Deepak, V Ghanavanth, Harish Bharadwaj R, R Uma, Gowranga Krishnamurthy. "Smart CCTV Surveillance System for Intrusion Detection With Live Streaming", 2018 3rd IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology

1%