

**SIX WEEKS ONLINE TRAINING REPORT**

**IN THE PARTIAL FULFILLMENT FOR THE AWARD OF THE DEGREE**

**OF**

**BACHELOR OF TECHNOLOGY**

**IN**

**COMPUTER SCIENCE AND ENGINEERING**

**Internship: E-box on object oriented programming using java**.

Submitted by:

Name of Student: Simran Manhas

Registration No: 11803523

Section: K18FR

**School of computer science & Engineering**

**Lovely Professional University, Phagwara**

( May-June, 2020).

**Acknowledgement**

I would like to express my special thanks of gratitude to my training website E-Box Online Training who gave me the golden opportunity to do this project on the topic “Object oriented programming using java”, which also helped me in doing a lot of work. I am really grateful to them.

Secondly I would like to thank my parents and friends who helped me a lot in finalizing this internship within limited time frame.

Last but not the least I would like to thank the almighty who gave me strength to complete my internship.

Simran Manhas

**TRAINING CERTIFICATE**

****

**Contents:**

1. Abstract

2. Introduction.

3. Technology learnt.

4. Reason for choosing Java.

5. Learning outcomes.

6. Conclusion.

7. Bibliography.

**Abstract**

Java is a class based object oriented programming language. Java was invented by the james gosling at Sun Microsystems in the year 1995. The history of java starts with the team named green team. Java is platform independent language. Java is the case sensitive language. Java works on the concept of (WORA) mean write once run anywhere. The syntax of java is similar to c and c++. My course is object oriented programming using java means using the oops concept in java they are classes, objects, collections, abstract, interface, exception handling, JDBC(java data base connectivity), inheritance and polymorphism, strings etc. object oriented programming using java helps us in creating different methods and variables and working with them and how to use the part of the code without compromising with security.

Object oriented programming concept in java helps to implement real word entities like polymorphism, inheritance, classes, objects, streams and writers, JDBC, collections, abstract, interface, exception handling etc. object oriented programming has many advantages it is faster and easier to execute and easier to maintain debug and modify object oriented programming helps us in reusing the code using inheritance property.

**INTRODUTION**

Java is a high level programming language. Java is a platform independent programming language. It was originally developed by the James Gosling at Sun Microsystems in the year 1995. Java works on the concept of WORA (write once and run anywhere). Java applications can be complied using byte codes. Among the programming languages present today java is one of the most popular language and it is also a client server web- application. Java was merged into the oracle corporation. Java language has most of the syntax same as C and C++ , but it has fewer low-level facilities than either of them. Some of the advantages of java are java is object oriented programming language. In java we can resuce the code using the property of inheritance in java. Java supports Classes, Object, Collections, inheritance and polymorphism, abstract and interface, strings, exception handling and multithreading. Java is very essay to learn and it does not require any basic knowledge before learning.

Java can be used for making many applications and applets. Java can be used for the applications like mobile, web based applications, gaming applications, desktop GUI (Graphical user interface) applications, scientific applications for creating (MATLAB), and java is used in big data as a backbone like hadoop, business applications, cloud based applications these are the important applications of java programming language and java is the case sensitive language. Java can also be used in the embedded systems like microchips and minicomputers to perform the dedicated tasks in the system. When we compare java 1.0 with java 1.1 is the library, Java 2.0 has larger library for handling the user interfaces.

The three important components of java are JVM( Java virtual machine) is a virtual machine and it makes the computer to run java programmes and other programmes with were written in other languages that are also complied to java byte code. JVM acts as run time engine for running the java applications. Java run time environment (JRE) is the software layer that runs on the top of the computer operating system software and provides class libraries and other resources that java program needs to run. Java development kit (JDK) is a important component of Java environment and JDK consists of java debugger, java classes and the other classes.

**Features of java:**

**1. simple**: Java’s developers deliberately left out many of the unnecessary features of other high-level programming languages. For example, java does not support pointer math, implicit type casting, structures or unions, operator overloading, templates, header files, or multiple inheritance.

**2. Object oriented programming:** Just like C++, Java uses classes to organize code into logical modules. At runtime, A program creates objects from the classes. Java classes can inherit from other classes, but multiple inheritance, wherein a class inherits methods and fields from more than one class, is not allowed.

**3. Statically typed:** In java all the objects used in a program must be declared before they are used. This enables the java complier to locate and report type conflicts.

**4. Complied :** Before you can run a program written in the java language, the program must be complied by the java complier. The compilation results in a “byte-code” file that, while similar to a machine-code file, can be executed under any operating systems that as a java interpreter. This interpreter reads in the byte-code file translates the byte-code commands in the machine language commands that can be directly executed by the machine that’s running the java program. You could say, then, that java is both a compiled and interpreted language.

**5. multithreading :** In the java programs contains the multiple threads of execution, which enables the program to handle several tasks concurrently. For example, a multithreaded program can render an image on the screen in one thread while continuing to accept keyboard input from the user in the main thread. All applications have at least one thread, which represents the program’s main path of execution.

**Object oriented programming:**

Java also supports object-oriented programming techniques that are based on a hierarchy of classes and well-defined and cooperating objects.

**Classes:** A class is a nothing but a structure that defines the data and the method to work on the data. Classes is a basic concept of object oriented programming which revolve around the real life entities and class is a user defined blue print or prototype from which objects are created. Classes in java defines a set of objects that shares a common structure and behaviour.

**Objects:** object is also a basic unit of object oriented programming and represents the real life entities. In java an instance is a synonym for a object. A newly created instance has data members and members and methods as defined by the class for that instance.

**Well-defined Boundaries and cooperation:** Class definitions should allow objects to cooperate during the time execution.

**Collections:** In java the collections is nothing but a framework that provides an architecture to store and manipulate the group of objects. In java collections can also achieve all the operations that you perform on a data such as sorting, searching, manipulation , insertion and deletion.

**Inheritance:** In java inheritance is nothing but an important pillar of object oriented programming. In java it is a mechanism by which one class is allow to inherit the features of another class. In java one object-oriented concept that helps the objects works together is inheritance and also inheritance defines relationships among classes in an object-oriented language. The relationship is one of parent to child where the child or extending class inherits all the attributes of the parent class. In java, all classes descend from java.lang.object and inherit its methods. In the inheritance the keywords is “extends”.

**Polymorphism:** In java polymorphism is nothing but the word poly means many and morphs means forms so it means many forms. In java we can define polymorphism as the ability of a message to displayed in more than one form. In java polymorphism is considered as one of the important features of object oriented programming. Polymorphism also allows us to perform a single action in different ways. In the other words, polymorphism allows you to define one interface and have multiple implementations. In java polymorphism is divided into two type one is compile time polymorphism and another one is run time polymorphism. In java compile time polymorphism is nothing but it is also known as static polymorphism. This type of polymorphism is achieved by function overloading or operator overloading. In java run time polymorphism is nothing but it also known as Dynamic method dispatch. It is a process in which a function call to the overridden methods is resolved at runtime. This type of polymorphism is achieved by method overriding.

**Abstract and Interface:** In java abstract is a keyword. In java abstract class does not support multiple inheritance and abstract class also contains a data member, constructors. In java an abstract class contains both incomplete (abstract) and complete member and only complete member of abstract class can be static. In java interface is a keyword and it supports multiple inheritance and it does not contain data member and constructors. In Java an interface contains only incomplete member and cannot have access modifiers by default everything is assumed as public and interface cannot be static.

**Strings:** String in java is basically an object that represents a sequence of char values. An array of characters works as a strings. In java arrays are immutable and strings are also immutable as well. Whenever a change to a string is made, an entirely new string is created.

**Multithreading:** In java multithreading is nothing but it is a java feature that allows the concurrent execution of two or more parts of a program for maximum utilization of CPU. In java each part of such program is called a thread. So, threads are light-weight processes within a process. In java threads can be created using two mechanisms one is Extending the thread class and another is implementing the runnable interface. We can create a class extends the java.lang.thread class. This class overrides the run() method available in the thread class. A thread begins its life inside run() method. We can create an object of our new class and call start() method to start the execution of a thread. Start() invokes the run() method on the thread object.

**Exception handling:** In java exception handling is used to communicate the existence of a run time problem or error from where it was detected to where it was detected to where the issues can be handled. It permits this is done in a uniform manner and separately from the main code, while detecting all errors, should an error occur, an exceptions is thrown(raised), which is then caught by the nearest suitable exception handler.

**Technology learnt:**

The technology I learnt is object oriented programming using java. In this course I have learnt how to write a program for a given problem. In this course I have learnt about problem solving skills for a given particular program.

In this course there are total 11 units. which was further divided into chapters and then topics so during my whole 6 week course I learned the following:

Introduction to object oriented programming using java

Chapter 1: Class and objects

* Classes and objects-learning resources.
* Constructors- learning resources.
* Static and final-learning resources.

Class: In java a class is a group of objects which have common properties. In Java class is a basic object oriented programming. In java the class is the specification or template of an object. A real-world example is circle. In java class is keyword and it is an member-variable.

Implementation with Example-Creating employee class:-

Import java.lang.reflect.constructor;

Import java.lang.reflect.InvocationTargetException;

Public class employee {

Private string name = “ Sujatha ”;

Private string company= “ ABC ”;

Public employee() {

Super();

}

Public employee(String name, string company) {

Super();

this.name = name;

this.company = company;

}

Public String getName() {

Return name;

}

Public void setName(String name) {

this.name = name;

}

Public String getcompany () {

Return company;

}

Public void setcompany (String company) {

this.company =company;

}

}

Object: In the java object is the real-time entity having some state and behaviour. In java, object is an instance of the class having the instance variables like state of the object and the methods as the behaviour of the object. In java Real world examples in object are dogs have state (name, color, breed, hungry) and behaviour (barking, fetching, wagging tail). Chair, bike, marker, pen, table, car, book, apple, bag, etc.

How to declare, create and initialize an object in java:-

Public class employee {

Private string name;

Private string company;

Public employee (String name, String company) {

Super();

this.name = name;

this.company = company;

}

Public String getName() {

Return name;

}

Public void setName(String name) {

This.name = name;

}

Public String getcompany() {

Return company;

}

Public void setcompany(String company) {

this.company = company;

}

Public static void main(String[] args) {

Employee employee = new employee (“venkat” ,”BVB”);

Employee employee2 = new employee (“praneeth”,”Gec”);

Employee employee3 = new employee (“ sai”,”IIT”);

Constructors in java: In java constructors is a special method that is used to initialize the methods. In java a constructor also contains collection of statements that are executed at time of object creation. In java a constructor is a block of codes similar to the method. It is called when an instance of the Class is created. At the time of calling constructor, memory for the object is allocated in the memory. It is a special type of method which is used to initialize the object. Every time an object is created using the new () keyword, at least one constructor is called. In java It calls a default constructor if there is no constructor available in the class. In such case, Java compiler provides a default constructor by default. There are two types of constructors in Java one is no-arg constructor, and another one is parameterized constructor.

**Note:** It is called constructor because it constructs the values at the time of object creation. It is not necessary to write a constructor for a class. It is because java compiler creates a default constructor if your class does not have any.

In java rules for creating java constructors are two rules for the constructor one is constructor name must be the same as its class name and another one is a constructor must have no explicit return type and a java constructor cannot be abstract, static, final, and synchronized.

There are two types constructors are there in java

1. default constructor also known as non argument constructor.

2. Parameterized constructor.

A constructor is also known as default constructor. When it does not have any parameter.

Syntax for default constructor:

< class\_name>(){}

Example for default constructor:

In this example, we are creating the non argument class in the car class.

Example:

Class car1 {

// creating a default constructor//

Car1(){

System.out.println(“ car is created”);

}

// calling a default constructor//

Car1 b = new car1();

}

}

2. Parameterized constructor:

A parameter which as specific number of parameters is called parameterized constructor.

Example for parameterised constructor:

class Student{

    int id;

    String name;

    Student(int i,String n){

    id = i;

    name = n;

    }

    void display(){System.out.println(id+ +name);}

   public static void main(String args[]){

Student s1= new Student(111,”Karan”);

Student s2= new student(222,”Aryan”);

S1.display();

S2.display();

}

}

Static and final :

Static keyword can be used in 4 ways:

1. Static variables.

2. Static methods.

3. Static blocks of code.

4. Static nested class.

1. Static variable: it is a variable which belongs to the class and not to object (instance). Static variables are initialized only once, at the start of the execution.

Final: The final keyword in java is used to restrict the user. The java final keyword can be used in many Context. Final can be of three types one is variable, second one is method, and third one is class.

The final keyword can be applied with the variables, a final variable that have no value it is called blank final variable or uninitialized final variable. It can be initialized in the constructor only. The blank final variable can be static also which will be initialized in the static block only. We will have detailed learning of these. Let’s first learn the basics of final keyword.

Chapter 2: collections-1

* Collection1-learning Resources.

The Collections in java is a frame work that provides an architecture to store and manipulate the groups of objects. Java collections can achieve all the operations that you perform on a data such as searching, sorting, insertion, manipulation, and deletion. Java collection means a single unit of objects. Java collection framework provides many interfaces (sets, list, queue, deque) and classes (array list, vector, linked list, priority Queue, Hash set, linked hash set, Tree set). A collections represents a single unit of objects.

Chapter 3: Relationship.

* Relationships-Aggregation-learning resources.
* Relationships-Assoication-Learning resources.

Chapter 4: Inheritance and polymorphism.

* Inheritance-learning resources.
* Polymorphism-learning resources.
* Method overloading-learning resources.

Inheritance: In java inheritance is nothing but an important pillar of object oriented programming. In java it is a mechanism by which one class is allow to inherit the features of another class. In java one object-oriented concept that helps the objects works together is inheritance and also inheritance defines relationships among classes in an object-oriented language. The relationship is one of parent to child where the child or extending class inherits all the attributes of the parent class. In java, all classes descend from java.lang.object and inherit its methods. In the inheritance the keywords is “extends”.

Polymorphism: In java polymorphism is nothing but the word poly means many and morphs means forms so it means many forms. In java we can define polymorphism as the ability of a message to displayed in more than one form. In java polymorphism is considered as one of the important features of object oriented programming. Polymorphism also allows us to perform a single action in different ways. In the other words, polymorphism allows you to define one interface and have multiple implementations. In java polymorphism is divided into two type one is compile time polymorphism and another one is run time polymorphism. In java compile time polymorphism is nothing but it is also known as static polymorphism. This type of polymorphism is achieved by function overloading or operator overloading. In java run time polymorphism is nothing but it also known as Dynamic method dispatch. It is a process in which a function call to the overridden methods is resolved at runtime. This type of polymorphism is achieved by method overriding.

Method overloading : In java method overloading means if a class has multiple methods having same name but different in parameters, it is known as method overloading. If we have to perform only operation, having same name of the methods increases the readability of the program. Suppose you have to perform addition of the given numbers but there can be any number of arguments, if you write the methods such as a(int,int) for there parameters then it may be difficult for you as well as other programmers to understand the behaviour of the method because its name differs.

Chapter 5: Abstract and interface.

* Abstract classes-learning resource.
* Interface-learning resource.

**Abstract and Interface:** In java abstract is a keyword. In java abstract class does not support multiple inheritance and abstract class also contains a data member, constructors. In java an abstract class contains both incomplete (abstract) and complete member and only complete member of abstract class can be static. In java interface is a keyword and it supports multiple inheritance and it does not contain data member and constructors. In Java an interface contains only incomplete member and cannot have access modifiers by default everything is assumed as public and interface cannot be static.

Chapter 6: Collections-2.

* Collections-2-learning resources.

Chapter 7: String/ String buffer/ String Tokenizer.

* String Introduction-learning resources.
* String Functions-learning resources.
* String Buffer / String Builder-learning resources.

String and string buffer class:

Both String and String buffer class contains the sequence of 16 bits uni code character.

String represents a sequence of character. The easiest way to represent a sequence of character in java is by using a char array.

Example:-

Char charArr[] = new char[4];

CharArr ={‘J’,’A’,’V’,’A’};

String buffer class:

An instance of java’s String Buffer Class represents a string that can be dynamically modified. String Buffer is a peer class of string class. While string class creates strings of fixed\_length, String buffer class creates strings with flexible length that can be modified in terms of both length and context. We can insert characters and substrings in the middle of a string or append another string to the end.

String Buffer class has two constructors.

StringBuffer() => 16 bits.

StringBuffer(int i)=> fixed by int value.

StringBuffer(“Str”);=> 16+str(length of str).

Chapter 8: Exception handling.

* Exception handling-learning resources.
* Try /catch block-learning resources.
* Custom exceptions learning resources.
* String and exception handling.

Exception handling: An exception is a condition that is caused by a run time error in the program, when the java interpreter encountered an error. It create an exception objects and throws it. In java Exception is an object that describes an exceptional conditions that has occurred in a piece of code. At the some points the exception is caught and processed is called Exception handling. In java exception can be generated by the JVM or they are manually generated.

In exceptional handling mechanism perform the following task and find the problem (hit the execution) and it inform that an error has occurred (throw the exception). In java it receive the error information (catch the exception) and take corrective action (handle the exception).

Types of exception or hierarchy of exception class:-

Exception is objects with hierarchical relationship. they have their own hierarchy. The root class of all the exception classes is the “throwable”.

Java.lang.\*;

Throwable :- A java exception is an instance of the class derived from Throwable. This class is contained in java.lang package and subclass of throwable contained in various packages.

Exception class :- This class describes the errors caused by programs. These errors can be caught and handled by the program. This is also the class that you will subclass to create our own custom exception types.

IO Exception :- This class describes errors related to input/output operation such as invalid input or inability to read from a file.

Error :- it is not in our control. The error class describe internal system errors, which rarely occur. Stack overflow is the best example of errors. Java exception handling is managed by five keywords are try, catch, finally, throw, throws.

Try-catch mechanism: in java keyword try to preface a block of code that is likely generate an error condition and ‘throw’ an exception. A catch block defined by catch handles it appropriate way.

The try block can have one or more than one statement that could generate an exception. If any one statement generate an exception, the remaining statement in the block are skipped and exception jumps to catch block.

Every try statement should be followed by at least one catch statement/ finally otherwise complication error will occur.

The catch statement is passed a single parameter, which is reference to the exception object thrown by the try block.

Chapter 9: JDBC (Java data base connectivity).

* JDBC-learning resources.
* JDBC API-learning resources.

JDBC In java: In java JDBC stands for java database connectivity. JDBC is a java API to connect and execute the query with the database. It is a part of java SE (java standard edition). JDBC API uses JDBC API uses JDBC drivers to connect with the database. There are four types of JDBC drivers:

1. JDBC-ODBC bridge driver,

2. Native driver,

3. Network protocol driver, and

4. Thin driver.

We have discussed the above four drivers in the next chapter.

We can use JDBC API to access tabular data stored in any relational database. By the help of JDBC API, we can save, update, delete and fetch data from the database. It is like open database connectivity (ODBC) provides by Microsoft.

Chapter 10: Multithreading.

* Multithreading-learning resources.
* Thread API-learning resources.

Multithreading in java: A typical program is executed sequentially and is single threaded. All modern operating systems may execute multiple programs simultaneously, even if there is only a single cpu available to run all the applications. Multithreading allows multiple task to execute concurrently within a single program. The main advantage of multiple threads can grab CPU time when one line of execution is blocked. By using multithreading, you can create programs showing animation, play, music, display.

Document and download files from the network simultaneously.

Java was designed from the start to accommodate multithreading. Synchronized method to avoid collision in which more than one thread attempt to modify the same object at the same time.

Thread model: Multithreading allows a running program to perform several tasks at the same time. Java users threads to enable the entire environment to be asynchronous. It makes maximum use of CPU because idle time (waiting time) can be kept to minimum. This helps to reduce the wastage of CPU life cycle.

The java runtime system depends on threads for many things and all class libraries are designed with multithreading in mind. A thread exists in several states.

Running Thread, Ready to run as soon as its gets CPU time, suspended thread, suspended thread can be resumed, thread can be blocked, thread can be terminated.

Thread: in java any single part of execution is called thread. The path is defined by elements such as stack and set of registers to store temporary memory. Every program has at least on thread. Each thread has its own stack, priority and virtual set of registers. Threads subdivided the runtime behaviour of programs into separate, independently running subtasks. Switching between threads is called out automatically by JVM. Even in a single processor system, thread provides an illusion of carrying out several tasks simultaneously.

Chapter 11: Streams and writers.

* Streams and writers-learning Resources.
* Files/learning resources.

Streams and writers: in java streams and writers is nothing but it is an abstract class for writing character streams. The methods that a subclass must implement are write( char[], int, int) , flush (), and close(). Most subclasses will override some of the methods defined here ro provide higher efficiency, functionality or both.

Chapter 12: Application development.

* Instasmart- Application development.

**Reason for choosing java:**

All of the above is the part of my training during my summer break I specially choose the object oriented programming using java by E-box for reasons stated below:

* **I was interested in programming in java since my first semester.**
* **Object oriented programming using java is a thing you need to know no matter in which language do you code.**
* **One need to learn how to make coding of a real life programs he/she is facing.**
* **It had video lectures of all the topics from which one can easily learn. I prefer learning from video rather than books and notes. I know books and notes and thesis have their own significance but still video lecture or face to face lectures make it easy to understand faster as we are involved Practically.**
* **It had 150+ coding programs with video explaind solutions.**
* **It had track based learning and weekly assesment to test my skills.**
* **In this platform there is a mentor also he explained a programs weekly three times after 5clock he is going to explain everything in that google meet class.**
* **It was a great opportunity for me to invest my time in learning instead of wasting it here and there during my summer break in this Covid-19 panademic situation.**
* **It contained a lot of knowledge for such a resonable price.**
* **This was a life time accessable course which I can use to learn even after my training whenever I want to revise.**
* **Along with all these reasons one of the reason was the E-box platform which is offering the course because E-box is one of the best platform for Computer Science Students.**

**Learning outcomes**

In this course I have learnt about object oriented programming using java. I have also learnt about topics like classes, objects, polymorphism, Inheritance, and how to implement them using java programming language. This course helped me to learn more about java.

I also learnt that java is a robust language as it is easily able to access exceptional handling and I also came to know that java can easily Provides memory management.

In this course I have learnt the multithreading which is one of the most improtant topic. I learnt about the JVM and its parts and it is very useful for me.

I have learnt all the important features in java like objects, classes, JBDC, streams and writers, abstract and interface, inheritance and polymorphism, exceptional handling, collections etc…

Through we are having high level languages like c and c++, but we are preferring java. I learnt the reason behind it and also came to know how java have to overcome those problems with c, c++.

I have learnt that java is a architecture neutral which is a problem with other high level languages. I learn the process of trading and able to code it.

I have also learnt about topics like classes, objects, polymorphism, Inheritance, exception handling, streams and writers, collections, JDBC, multithreading, abstract and interface, strings and string buffer, and how to implement them using java programming language.

In this course I have learnt about classes is a nothing but a structure that defines the data and the method to work on the data. Classes is a basic concept of object oriented programming which revolve around the real life entities and class is a user defined blue print or prototype from which objects are created. Classes in java defines a set of objects that shares a common structure and behaviour.

In this course I have learnt about object is also a basic unit of object oriented programming and represents the real life entities. In java an instance is a synonym for a object. A newly created instance has data members and members and methods as defined by the class for that instance.

In this course I have learnt about inheritance is nothing but an important pillar of object oriented programming. In java it is a mechanism by which one class is allow to inherit the features of another class. In java one object-oriented concept that helps the objects works together is inheritance and also inheritance defines relationships among classes in an object-oriented language. The relationship is one of parent to child where the child or extending class inherits all the attributes of the parent class. In java, all classes descend from java.lang.object and inherit its methods. In the inheritance the keywords is “extends”.

In this course I have learnt about abstract and interface in java abstract is a keyword. In java abstract class does not support multiple inheritance and abstract class also contains a data member, constructors. In java an abstract class contains both incomplete (abstract) and complete member and only complete member of abstract class can be static. In java interface is a keyword and it supports multiple inheritance and it does not contain data member and constructors. In Java an interface contains only incomplete member and cannot have access modifiers by default everything is assumed as public and interface cannot be static.

In this I have learnt many things of java and it is very interesting to learn the things.

**Thank you.**