Final Report

Question	Explanation
1	What is a main method in Java ? Main method in Java is a standard method which is used by JVM to start execution of any Java program. Valid Signature of main method in Java Main method is a standard method and has pre specified signature, if you change the signature of main method JVM will not be able to locate main method will throw Exception at runtime as shown in above example. ▪ public static void main(String[] args) ▪ public static void main(String []args) ▪ public static void main(String args[]) ▪ public static void main(String... args) ▪ static public void main(String[] args) ▪ public static final void main(String[] args) ▪ final public static void main(String[] args) ▪ final strictfp public static void main(String[] args) Why main method is public static and void in Java ? Main method in Java is public so that its visible to every other class, even which are not part of its package. If its not public JVM classes might not able to Access it. Main method is static in Java, so that it can be called without creating any instance. While JVM tries to execute Java program it does not know how to create instance of main class as there is no standard constructor is defined for main class. Main method is void in Java because it doesn’t return anything to caller which is JVM.
2	What is object oriented programming language ? Object Oriented programming is a programming style which is associated with the concepts like class, object, Inheritance, Encapsulation, Abstraction, Polymorphism. Inheritance is one such concept where the properties of one class can be inherited by the other. It helps to reuse the code and establish a relationship between different classes. Encapsulation is a mechanism where you bind your data and code together as a single unit. It also means to hide your data in order to make it safe from any modification. Abstraction refers to the quality of dealing with ideas rather than events. It basically deals with hiding the details and showing the essential things to the user. Polymorphism means taking many forms, where ‘poly’ means many and ‘morph’ means forms. It is the ability of a variable, function or object to take on multiple forms.
3	What is bytecode in Java ? Java bytecode is the instruction set for the Java Virtual Machine. It acts similar to an assembler which is an alias representation of a C++ code. As soon as a java program is compiled, java bytecode is generated. In more apt terms, java bytecode is the machine code in the form of a .class file. With the help of java bytecode we achieve platform independence in java.
4	What is java.util package ? It contains the collections framework, legacy collection classes, event model, date and time facilities, internationalization, and miscellaneous utility classes (a string tokenizer, a random-number generator, and a bit array). What is java.sql package ? The java.sql package contains the entire JDBC API that sends SQL (Structured Query Language) statements to relational databases and retrieves the results of executing those SQL statements.
5	Key features of the Object Oriented programming - Emphasis on data rather than procedure

	- Programs are divided into entities known as objects - Data Structures are designed such that they characterize objects - Functions that operate on data of an object are tied together in data structures - Data is hidden and cannot be accessed by external functions - Objects communicate with each other through functions - New data and functions can be easily added whenever necessary - Follows bottom up design in program design
6	What is local variable in Java ? A local variable in Java is a variable that's declared within the body of a method. Then you can use the variable only within that method. Other methods in the class aren't even aware that the variable exists.
7	What is Java Package ? A java package is a group of similar types of classes, interfaces and sub-packages. Package in java can be categorized in two form, built-in package and user-defined package. There are many built-in packages such as java, lang, awt, javax, swing, net, io, util, sql etc.
8	What is code comment ? The Java comments are the statements that are not executed by the compiler and interpreter. The comments can be used to provide information or explanation about the variable, method, class or any statement. It can also be used to hide program code.
9	What is java class file ? A Java class file is a file (with the .class filename extension) containing Java bytecode that can be executed on the Java Virtual Machine (JVM). A Java class file is usually produced by a Java compiler from Java programming language source files (.java files) containing Java classes (alternatively, other JVM languages can also be used to create class files). If a source file has more than one class, each class is compiled into a separate class file. JVMs are available for many platforms, and a class file compiled on one platform will execute on a JVM of another platform. This makes Java applications platform-independent.
10	Java Variables A variable is a container which holds the value while the Java program is executed. A variable is assigned with a data type. Variable is a name of memory location. There are three types of variables in java: local, instance and static. There are two types of data types in Java: primitive and non-primitive. A variable declared inside the body of the method is called local variable. You can use this variable only within that method and the other methods in the class aren't even aware that the variable exists. A variable declared inside the class but outside the body of the method, is called instance variable. It is not declared as static. A variable which is declared as static is called static variable. It cannot be local. You can create a single copy of static variable and share among all the instances of the class. Memory allocation for static variable happens only once when the class is loaded in the memory.
11	Java Import Statements In Java, the import statement is used to bring certain classes or the entire packages, into visibility. As soon as imported, a class can be referred to directly by using only its name. The import statement is a convenience to the programmer and is not technically needed to write complete Java program. If you are going to refer to some few dozen classes into your application, the import statement will save a lot of time and typing also. In a Java source file, the import statements occur immediately following the package statement (if exists) and before any class definitions.
12	Differences between public, protected, default and private class in Java public - The class is accessible in any package in the program. protected - The class is accessible in the package and in subclasses of the class. default - Accessible only in the package where the class is defined. private - Cannot be accessed from an external class at all even if they are in same package or different package.
13	Notable features of Java

	Object Oriented: In Java, everything is an Object. Java can be easily extended since it is based on the Object model. Platform Independent: Unlike many other programming languages including C and C++, when Java is compiled, it is not compiled into platform specific machine, rather into platform-independent byte code. This byte code is distributed over the web and interpreted by the Virtual Machine (JVM) on whichever platform it is being run on. Simple: Java is designed to be easy to learn. If you understand the basic concept of OOP Java, it would be easy to master. Secure: With Java's secure feature it enables to develop virus-free, tamper-free systems. Authentication techniques are based on public-key encryption. Architecture-neutral: Java compiler generates an architecture-neutral object file format, which makes the compiled code executable on many processors, with the presence of Java runtime system. Portable: Being architecture-neutral and having no implementation dependent aspects of the specification makes Java portable. The compiler in Java is written in ANSI C with a clean portability boundary, which is a POSIX subset. Robust: Java makes an effort to eliminate error-prone situations by emphasizing mainly on compile time error checking and runtime checking. Multithreaded: With Java's multithreaded feature it is possible to write programs that can perform many tasks simultaneously. This design feature allows the developers to construct interactive applications that can run smoothly. Interpreted: Java byte code is translated on the fly to native machine instructions and is not stored anywhere. The development process is more rapid and analytical since the linking is an incremental and light-weight process. High Performance: With the use of Just-In-Time compilers, Java enables high performance. Distributed: Java is designed for the distributed environment of the internet. Dynamic: Java is considered to be more dynamic than C or C++ since it is designed to adapt to an evolving environment. Java programs can carry an extensive amount of run-time information that can be used to verify and resolve accesses to objects at run-time.
14	First line of code in Java The first line defines a class called Main. In Java, every line of code that can actually run needs to be inside a class. This line declares a class named Main, which is public that means that any other class can access it.
15	Using Package Members Only public package members are accessible outside the package in which they are defined. To use a public package member from outside its package, you must do one or more of the following: ▪ Refer to the member by its long (qualified) name ▪ Import the package member ▪ Import the member's entire package Each is appropriate for different situations, as explained in the following sections.
16	What is difference between javac and java commands? The javac command is used to compile Java programs, it takes .java file as input and produces bytecode. The java command is used to execute the bytecode of java. It takes byte code as input and runs it and produces the output.
17	What is encapsulation ? Encapsulation is one of the four fundamental OOP concepts. Encapsulation in Java is a mechanism of wrapping the data (variables) and code acting on the data (methods) together as a single unit. In encapsulation, the variables of a class will be hidden from other classes, and can be accessed only through the methods of their current class. Therefore, it is also known as data hiding.
18	Count Operator Java uses a simple operator called the increment operator to streamline the process of adding one to any integer variable. The increment operator adds on to the value stored in the variable and stores the result within the variable.

19	Bytecode vs Machine Code The main difference between the machine code and the bytecode is that the machine code is a set of instructions in machine language or binary which can be directly executed by the CPU. While the bytecode is a non-runnable code generated by compiling a source code that relies on an interpreter to get executed.
20	A semicolon(;) is the character for terminating a statement in Java.
21	Private is a Java keyword which declares a member's access as private. That is, the member is only visible within the class, not from any other class (including subclasses). The visibility of private members extends to nested classes.
22	Classes in Java 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 can include modifiers, class name, superclass(if any), interfaces(if any) and body. Constructors are used for initializing new objects. Fields are variables that provide the state of the class and its objects, and methods are used to implement the behavior of the class and its objects.
23	The meaning of platform-independent is that the java compiled code(byte code) can run on all operating systems.
24	What is JVM ? Java Virtual Machine (JVM) is a engine that provides runtime environment to drive the Java Code or applications. It converts Java bytecode into machines language. JVM is a part of Java Run Environment (JRE). In other programming languages, the compiler produces machine code for a particular system. However, Java compiler produces code for a Virtual Machine known as Java Virtual Machine.
25	Class variables are always in scope for the entire program.
26	Two types of “import” statement in Java 1. Single Type Import: "import type_name;" - It tells JVM to load the definition of a single type immediately from the specified package. 2. On-Demand Type Import: "import package_name.*;" - It tells JVM to search the specified package for any missing type when it is needed.
27	Correct order of statements for a Java class file: 1- Package statement 2- Import statements 3- Class declaration
28	Java Packages: Built-in and User defined Built-in Package: Existing Java Package for example java.lang, java.util etc. User-defined-package: Java package created by user to categorize their project’s classes and interface.
29	Print() in Java Print method in Java is used to display a text on the console. This text is passed as the parameter to this method in the form of String. This method prints the text on the console and the cursor remains at the end of the text at the console.
30	The javac command compiles a .java file into .class bytecode file.
31	Overloading in Java Overloading allows different methods to have the same name, but different signatures where the signature can differ by the number of input parameters or type of input parameters or both. Overloading is related to compile-time (or static) polymorphism.
32	Data types are divided into two groups: • Primitive data types - includes byte, short, int, long, float, double, boolean and char • Non-primitive data types - such as String, Arrays and Classes
33	Java println() method The println() method is similar to print() method except that it moves the cursor to the next line after printing the result. It is used when you want the result in two separate lines. It is called with “out” object. If we want the result in two separate lines, then we should use the

	println() method. It is also an overloaded method of PrintStream class. It throws the cursor to the next line after displaying the result.
34	One of the most important reason Java supports inheritance to allow increased code reuse among classes. Developers minimize duplicate code in new classes by sharing code in a common parent class.
35	Java Comments Comments can be used to explain Java code, and to make it more readable. It can also be used to prevent execution when testing alternative code. Single-line comments start with two forward slashes (//). Multi-line comments start with /* and ends with */.
36	Valid declaration of an entry point in Java application public static void main(String… arguments) public static final void main(String[] arguments) public static void main(String[] arguments)
37	Java Inner Class Any non-static nested class is known as inner class in java. Java inner class is associated with the object of the class and they can access all the variables and methods of the outer class. Since inner classes are associated with the instance, we can’t have any static variables in them. The object of java inner class is part of the outer class object and to create an instance of the inner class, we first need to create an instance of outer class.
38	A class declaration is required to define a valid Java class file.
39	.java is the proper filename extension for a Java source file.
40	Java.lang package in Java Provides classes that are fundamental to the design of the Java programming language. The most important classes are Object, which is the root of the class hierarchy, and class, instances of which represent classes at run time.
41	Important packages in Java **java.lang:** This package got primary classes and interfaces essential for Java program. It consists of wrapper classes (wrapper classes can be used to convert ordinary data into objects), Strings, Threads etc. **java.util:** This package contains useful classes and interfaces like Stack, LinkedList, Arrays, ArrayList, List etc... **java.io:** This package handles files and input output related tasks. **java.awt:** This package helps to develop GUI. It consists of an important sub package java.awt.event to handle the events for GUI elements. **java.swing:** This package helps to develop GUI like java.awt. In fact, this an extension to java.awt. **java.net:** Client - server programming can be done using this package. This is a very important package to develop any web program like web browse or web server. **java.applet:** Applets are programs which come from a server into a client and get executed on the client machine. This package was very well known to everyone before the development of servlets. **java.sql:** This package helps us to connect to database like Oracle.
42	Object-oriented programming is the technique of structuring data into objects, which may contain data and a set of actions that operate on the data.
43	If the required package removes, the class does not compile so necessary package should be applied or it should not be removed.
44	Static Boolean FALSE: The Boolean object corresponding to the primitive value false. Static Boolean TRUE: The Boolean object corresponding to the primitive value true. Static Class: The class object representing the primitive type Boolean.
45	A Java method is a collection of statements that are grouped together to perform an operation.

46	Creating Method
	Public static int methodName(int a, int b) {
	//body

	} Here, ▪ public static – modifier ▪ int – return type ▪ methodName – name of the method ▪ a, b – formal parameters ▪ int a, int b – list of parameters Method definition consists of a method header and a method body.
47	Java may use a period . to separate packages and javac takes a .java file and returns a .class file.
48	What is an Exception ? An exception is an unwanted or unexpected event, which occurs during the execution of a program i.e at run time, that disrupts the normal flow of the program's instructions. Error vs Exception Error: An Error indicates serious problem that a reasonable application should not try to catch. Exception: Exception indicates conditions that a reasonable application might try to catch.
49	When we define the class, we also should define the attributes and methods.
50	Garbage collection can happen at any time while an application is running, especially if the available memory suddenly becomes low. It is trivial to create a Java application with an infinite loop that never terminates. The computer must be able to run the JVM in order to execute a Java class. The JVM does require an entry point method to begin executing the application.