Homework 1 Report

Q1: In Java programming language, every application must contain a main method. This method predefined in JVM to indicate as a start point. Java Virtual Machine starts execution by invoking the main( ) method to reach some specified class. To make it work properly, there are some appropriated rules to write main( ) method:

- Main method includes **static** keyword. This means that the method starts without creating the instance. When JVM starts, it does not create any object for using the main method. Absence of static keyword gives an error in the console.

- The main method accepts some data from the user. It accepts an array of elements by type **String**. When there is a lack of **String[ ]** in java program, it will compile but not run, because JVM will not recognize the main method. JVM always looks for the main method with a string type array as a parameter.

- Main method should be accessible from anywhere. So, we should use a **public** keyword before the main( ) method so that Java Virtual Machine can specify the execution point of the program. Main method should be visible to JVM.

- This main method includes void return type. There is **final** keyword in option. This means that the main method cannot overridden. In addition, this public method have a static property. It provides this method accessed without creating the instance. This method has a parameter that is a String array called mydata –array name may change-. There is everything needed for this method to be a valid.

Answer is Option D

Q2: Silver class has the luster **attribute**. Gold and Silver classes are same level and them both subclasses. Moreover, Luster attribute does not inherit.

According to the given class diagram, Metal class is the **parent class** of Gold and Silver classes. This means that some data elements and functions inherited from the parent class. The child class have some extra elements and functions. *Weight* and *Color* properties inherited from parent class to the child ones.

Java is an object oriented programming language and one of the feature is **inheritance.** The diagram tells us about inheritance feature of Java.

The meaning of platform independence is that java compiled code (byte-code) can run on different operating systems. There is no demonstration of **platform independence** in this diagram.

Answer is Option A

Q3: Java source code files given a name ending with .**java** extension.

When a java program compiled, the file produced by the computer ends with **.class** file extension

The file produced by the Java Compiler contains **bytecode** that executed by the JVM.

.dll file extension (Dynamic Link Library) is a complied library that contains a set of procedures and drivers are executed by Windows program.

Answer is Option C

Q4: The code does not compile because of line 4.

I try to compile but the error message is "The type Date is ambiguous"

Answer is Option B

Q5: **Classes** defined as a group of objects that have common properties.

A **subclass** object can assigned as a super class object. This process known as object casting. Casting can make from **superclass** to the **subclass** either. Thus, an object takes different forms.

When the **object** created, system allocates memory for new object and return a reference to that memory. After initialization that means giving initial values deployed by a **constructor**, objects can hold the data.

A **method** is a collection of statements that perform an operation and return a result to the caller. The aim of methods define the code once and use it many times.

Answer is Option A

Q6: Java **interface variables** are available at all places of the program.

Class variables are also known as static variables. These variable become accessible from in class

Instance variables bound to their instances. These variables are visible for methods, constructors in the class block.

Local variables declared inside a method that have method level scope and cannot accessed outside of the method.

Answer is Option D

Q7: Java package is a group of similar types of classes, interfaces and sub packages.

Java.util package contains the collections framework, legacy collection classes, event model, date and time facilities and another various utility classes

Java.lang package provides classes that are fundamental to the design of the Java Programming language. Because of **java.lang** package so essential, this package implicitly imported by every Java source file.

Java.lang.system class contains several useful classes It contains standard input output streams, loading libraries and access to external properties. It cannot be instantiated. This facilities provided by system.

There are no system.lang and java.system packages in java classes.

Answer is Option B

Q8: The Java language supports three types of comments.

/* comments */	The complier ignores everything from /* to */.
// comment	The compiler ignores from // to the end of line.
/** documentation */	This is a documentation comment and in general its called.
#	This sign is a comment symbol for linux terminal.

Answer is Option C

Q9: A Java file consist of none or more **import** packages, an optional **package** directive, one or more **class** definitions. In Java files, each file should contain at most one class that declared **public**. A public class designed for use by other classes and packages.

Answer is Option D

Q10: Answer is Option B

Q11: Import declaration helps the compiler to locate a class for used in this program.

Unused import statements will not cause to compilation to fail. IDE's just warning you about unused import statements. Duplicate import statement is not acceptable in many authorities.

Java complier looks for the classes imported in the classpath. If they are not there in classpath then class not found exceptions thrown.

If a class not contain a import statement that what you need, it cannot compile because compiler must know how the manipulate the given class.

Answer is Option B

Q12: This class is not compiled. **The answer is Option A**

Q13: ***java*** is the Java application launcher tool for executing programs in written Java language an compiled into **bytecode**. The ***java*** command can execute the class file with definition of classpath given as an argument. (First claim is False)

Java is the object oriented programming language. (Second claim is False)

javac command reads java source file (.java) and compiles them into bytecode class files (.class) that run on the Java Virtual Machine (Third claim is False)

The answer is option D

Q14: **import** widget.*; Import commands used mostly top of the lines.

// **widget** Manager; Comment line usable everywhere.

package sprockets; Package definitions are allowed in first line.

int facilityNumber; This utilization is not allowed in first lines of Java

Answer is Option D

Q15: If you not even identified package statement, the class in the default package. But every class don't need to package declaration.

To create a package, this syntax used: **package** myPackage

Classes in packages can have fields and methods that are visible by all classes inside a package, but not outside.

In oppose of this option, access restriction can applied to object and methods with use of packages.

Answer is Option C

Q16: To compile a program execute the java compiler by name of file for this question

javac Manager.java

Java compiler creates a class file. And then, execution of the application

Java Manager

Answer is Option B

Q17: Structuring a Java class such that only methods with in the class can access its instance variables referred to as encapsulation.

Answer is Option D

Q18: This code snippet does not compile. **Answer is Option D**

Q19: JVM takes bytecode and compile with the Just in Time compiler. Compiler convert the bytecode harmoniously. Java is not a compiler type language. Java has a philosophy that write once run everywhere. When looking inside of bytecode, it contains meaningless characters at first time. Then a hexedit program make this file understandable for machine. This program not allow changing the content.

Answer is Option A

Q20: In Java programming language, a semicolon used at the end of each statement.

Answer is Option D

Q21: After the compiling and executing, the class tomorrow variable comes from the out of main scope. Then we reach 20 + 10 + 1 = 31 as a result.

Answer is Option C

Q22: There is no compilation error in Line 3 only. **Answer is Option C**

Q23: A java **class** can run on many digital devices with the platform independence feature of Java.

Answer is Option D

Q24: Java Virtual Machine has garbage collector for unused fields. Therefore, it is responsible for memory management. JVM is available for every operating system. Hence, it supports the platform independence. JVM takes java instructions then to turning into bytecode not machine instructions.

Answer is Option D

Q25: Class variables also known as **static** variables are always usable in scope for entire program.
Answer is Option B

Q26: Wildcard symbol (*) means that only **import** classes form in given package, not from the any sub-packages. In many IDE there is an option for organizing imports. When writing an **import** statement we pay attention to how the packages written. Only option C is included properly.

Answer is Option C

Q27: Correct order of statements in Java class file is
package statement – **import** statement -- **class** declaration.

Answer is Option D

Q28: There are three of import statements can be discarded.

import java.lang.* **import** java.lang.Object **import** stars.*

Answer is Option D

Q29: Ellipsis notation (…) also known as varargs any number of argument can added as a parameter. This can placed end of the method signature. In question string… is used this means that there is an array with string type. Arguments processed sequentially. In addition, expressions with quotation marks counted as one. Only third option white-tailed displays as a message.

Answer is Option D

Q30: The **javac** command is abbreviation of java compiler compiles a java class (.java) file into the **bytecode** (.class) file.

Answer is Option B

Q31: Method overloading means a method can have same name but it differ from number of parameter & type of parameter. We cannot define operator overloading in java. Procedural programming language allows the calling the functions at any point. Java is not accounted for procedural language.

Answer is Option B

Q32: This code snippet start with a package agent. Then, there must be a class expression in java. A method can uses long primitive data type.

Answer is Option D

Q33: The output of this application is 2 5. **Answer is Option A**

Q34: Inheritance in java that when we try to inherit a class from current class, we can use methods and data of **parent** class. This provides the less duplicate of code in new classes created.

Answer is Option D

Q35: We can use multiple slash character (/) in comment lines in java.

Answer is Option A

Q36: The main method can take the **string** parameter specified with three dot. Moreover, it may be have some keywords like **final**. However, there must be a string array in argument section.

Answer is Option B

Q37: **public** String color statement not placed outside of class definition. No global variables allowed in Java. Then when the method has **access modifier** It involves into method scope. We don't need to use same modifier for data fields.

Answer is Option B

Q38: A class declaration is required to define a valid Java class file.

Answer is Option A

Q39: .java is the source file extension. **Answer is Option D**

Q40: In give class, Math classes are collide with **java.lang** and **pocket.complex** packages. Therefore, this code does not compile at the import statement.

Answer is Option A

Q41: All of the import statements give the access of only one degree below in Java. **import dog.*** and **import dog.puppy** given in the question. In A Option **dog.puppy.female.KC** we reach the female class only not KC.

Answer is Option A

Q42: Object orientation means using objects in programming (Just like Java). It has attributes for hold the data and consist of actions to use the object with classes or other objects.

Answer is Option B

Q43: In given class definition all import statements connected with a class. Then whatever we discard the import statement we have code compilation problem.

Answer is Option A

Q44: There is compile error in this application. Don't throw an exception about this error.

Answer is Option C

Q45: Code compiled. It takes the wheel menu from q object. Result is 20.

Answer is Option D

Q46: The class is compiled. Color variable takes the nearest value (in local value).

Answer is Option B

Q47: **java** command use a period (dot) to separate packages.

Javac takes the java class file (.java) and return it to the bytecode file (.class)

Answer is Option C

Q48: This application not compiled because compiler does not see the main method and give **the** error message. Message content is **: "** `Main method not found in class hw_1.Q48,` **"**

Answer is Option C

Q49: Class diagram provide the visualizing the classes on the project. A class is represented as a box with 3 compartments. The uppermost one contains the **class name**. The middle one contains the class **attributes** and last one contains the class **methods**.

Class members (attributes and methods) have a visibility option given them. Its represented by one character.

public (+) **private (-)** **protected (#)** **package (~)**

According to this information, there is a class named **Book**, Book class have a **public** attribute called **numberOfPages**, and this class has a **public getRating()** method in given class diagram.

Resource : https://medium.com/@smagid_allThings/uml-class-diagrams

Answer is Option C

Q50: **JVM** (Java Virtual Machine) acts like a runtime engine to Java Application. JVM actually calls the **main** method present in a java code. JVM is a part of **JRE(Java Runtime Environment)**. JVM does not schedule the garbage collector. JVM never ensure the application will always terminate. Java compiled code cannot able to run any computer.

Answer is Option C