

MidTerm Assignment

Fall 2020-21

Saiyur Rahman
ID: 19-39890-1

Course: Object Oriented Programming (1)

Section: L

Submitted to: MD. NAZMUL HOSSAIN
Lecturer, CS

Answer of Question No: 1

Briefly Explanation of following keyword

- a OOP: Full meaning of OOP is object oriented programming. Which means it create objects and you can contain data and code.
- b Class: Class in java is simillar type of building Blueprint. Which contains objects.
- c Object: An animals has legs, hair, ears object one same animal is a class and legs, hair are object. If we say many objects make a class.
- d Portability: Java is a multiplatform language so if once we built a programme we can easily use it in Windows OS and Mac OS & Linux Main is we can carry the Java bytecode.
- e Robustness: Java is so ~~securid~~ secured and easily usab useable they have 3 type of encapsulation and extra default method so no one can easily access at code with Strong memory management, automatic garbage collection by JVM and have Exception handling.

f Multithreaded:

Java is a multi-threaded programming language which means we can develop multi-threaded program using Java. A multi-threaded program contain two or more parts that can run concurrently.

g Javac: Javac means Java Compiler who compile the code into computer.

h Literals: A literal is a source code representation of a fixed value. They are represented directly in the code without any computation.

For Example

int num = 10;

Data Type

Literal

i Default: Java has a "default" type word which is silently looks same as none. For Example

```
default class MyName{  
}
```

and

```
class MyName{  
}
```

both are same.

j **new**: new is mainly use in oop concept when we need to create any thing like object need to allocated memory so new keyword use for allocated memory in computer.

k **this**: this is a reference variable in Java that refers to the current object. It can be used to refer instance.

l **Scanner**: Scanner is a class. It's use for get user input and it is found in the java.util package.

m **compareTo()** : compareTo() is a String class method which are used to compare a string to another string.

n **capacity()** : capacity() is also a string method but mainly use in StringBuilder & StringBuffer class. It's returned the capacity on amount of available storage.

Answer to the Question no: 2

Q "Java supports multithreaded programming"

By definition, we can say that when any program run multitasking on multiple application it's need multi-thread. A multithread program contains two or more part's that can run concurrently and each parts can handle a different task at the same time making optimal use of the available resource specially when our computer has multiple CPUs.

Java Supports multithreaded Programming language
 which means we can develop multi-threaded program on application by using java.

For Example:

① In java there has a interface which is need to implement ① class MyProgram implements Runnable
 ② that interface has run() method which will run multithread ② public void run () ————— ②
 ③ Interface also have a class constructor named Thread ③ Thread (Runnable Obj , String threadName);
 ④ To start multi-threaded program need to use start method ④ void start (); ————— ④

Java is best known for its security. We can develop virus free systems. Java is secured because

- ① No explicit pointer
- ② Classloader in Java of the Java Runtime Environment (JRE) which is used to load Java classes into the Java Virtual Machine (JVM) dynamically. It adds security by separating the package for the classes of the local file system from those that are imported from network sources.
- ③ Byte-code Verifier: It determines what it checks the code fragments for illegal code that can violate access to objects.
- ④ Security Manager: It determines what resource a class can access such as reading and writing to the local disk.

If we compare the full coding process of java and C++

~~for Java~~ → Java (source code) → compiler → compile to bytecode (.class file) → JVM → convert/execute to binary (01001) → OS → execute

~~for C++~~ → C++ (source code) → compiler → compile to binary → OS → execute.

C

Java is platform independent because it is different from other language like C, C++, etc. Which are compiled into platform specific machines while Java is a write once, run anywhere language. A platform is the hardware or software environment in which a program runs. Java provides software based platform. It has two components

① Runtime environment

② API (Application Programming Interface)

And for these byte code conversions it can run anywhere.



100% Objected Oriented Language/Programming need

① Encapsulation/Data hiding

② Inheritance

③ Polymorphism

④ Abstraction

⑤ All predefined types are object

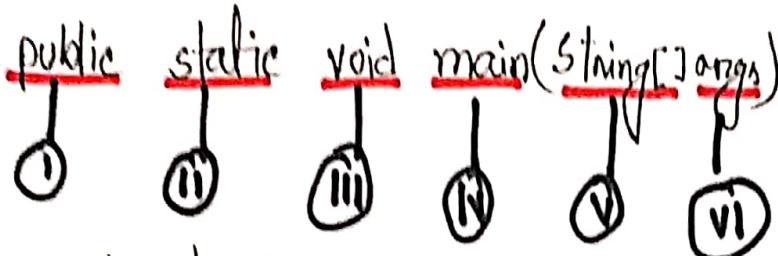
⑥ All users defined types are object

⑦ All operations performed on the object must be only through methods exposed at the object.

Java follows on they has

① Predefined primitive data types (which are not object)

② We can access the members of static class without creating object.



Every Word I mean ① to ⑥ all has a meaning to the JVM (Java Virtual Machine)

① Public: It is an Access modifier in Java there are 4 type of Access modifier

① Public

② Private

③ Protected

④ Default → ① Special for Java

public modifier is open for all anyone can access the method, Making it available for everyone the main() method that's why it's used. It is also made public so that JVM can invoke it from outside the class.

② Static: It is a keyword which is when associated with method, makes it class related method. The main() method is static so that JVM can invoke it without instantiating the class. This also save unnecessary wastage of memory which would have been used by the object declared only for calling the main() method by the JVM.

⑩ Void: It is also a keyword and used to specify that a method doesn't return anything. As main() method doesn't return anything, its return type is void.

⑪ main: It is the name of Java main method. It is the identifier that JVM looks for as the starting point of the java program. It's not a keyword.

⑫ String: It stores an array of type which are package in `java.lang.String`.

⑬ args: It is a part of String but it stores Java command line arguments. Here the name `Strings args` means array arguments but it is not fixed and user can use any name place it.

f
Java main method is always static, so that compiler can call it without the creation of an object or before creation of an object of the class. In any Java program program, the main() method is the starting point from where compiler starts program execution. So, the compiler needs to call the main() method. If the main() is allowed to be non-static then while calling the main() method JVM has to instantiate its class. The main method() must declared in public, static, void otherwise compiler show.

Q

Yes, We can execute a java program without a main method by using a static block.

Static block in Java is a group of statements that gets executed only once when the class is loaded into the memory by Java ClassLoader. It is also known static initialize block. Static initialization block is going directly into the stack memory. For example-

```
public class staticBlockMethod {
```

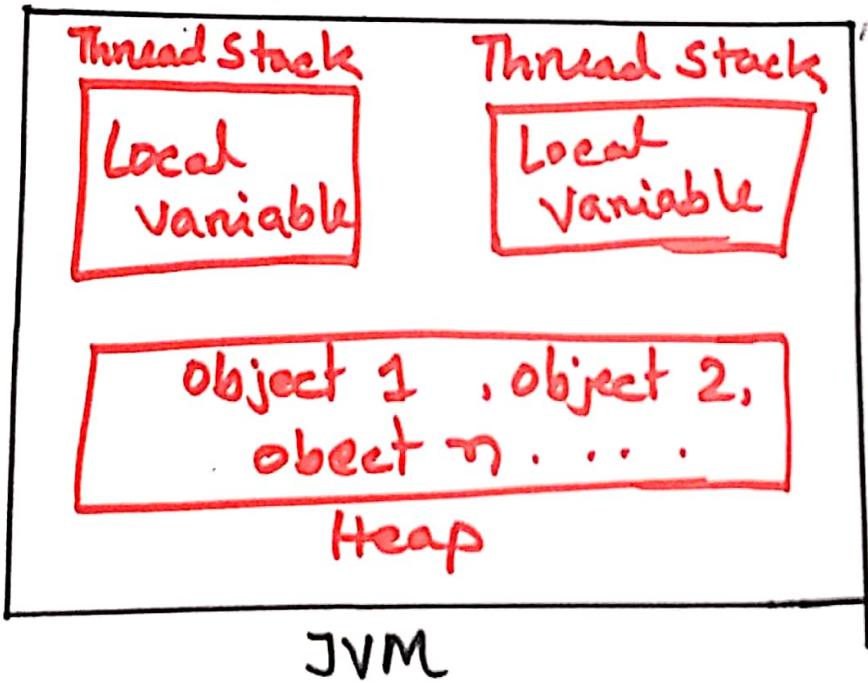
```
    ① - static {  
        ② -     System.out.println("Execute Java  
        ③ -     programming without main method");  
        ④ - }  
    }
```

From line ① to ④ is the method how can we make an initialization of static block by Static keyword.

Note: From Java 7 and newer versions don't allow this because JVM checks the presence of the main method before initializing the class.

In Java heap are used for memory storage. Java memory model is divided between Thread Stacks (One for each Thread) and a heap area.

① Thread Stack: It is a thread specific memory area and contains local variables, methods call information etc. JVM stacks could be of fixed size or variable size. If computation in a thread exceeds it's stack size limit then JVM throws StackOverflowError and exits.



i Parsing or Parse Parse is to read the value of one object to convert it to another type. When we enter some keywords in a search engine they parse the keywords and give back the results by searching for each word. In Java there are many uses of parse some of them are:

① index of which are used to calculate the position of that string and return a Integer value. We can say that convert string to Int.

② parseInt, parseFloat , etc .

By coding Representation

indexof()

```
public class ParseIndexof {  
    public static void main(String[] args) {  
        String s = "I am Saifydun"  
        int i = s.indexOf("S");  
        System.out.println(i);  
    }  
}
```

Show's 5 which is integer

parseInt

```
public class ParseInteger {  
    public static void main(String[] args) {  
        String s = "1234"  
        int i = Integer.parseInt(s)  
        System.out.println(i);  
    }  
}
```

Show's 1234 but not in String

We all know that java is not pure object oriented programming because of primitive data's int, float, double, long long, boolean. To make java pure object oriented language there have a class which are wrap that primitive data's and shows that data as a object type. The class is called wrapper class. That wrapper classes are

Primitive Type	Wrapper class
boolean	Boolean
char	Character
byte	Byte
short	Short
int	Integer
long	Long
float	Float
double	Double

~~K~~ No, we cannot define a static constructor in Java. If we are trying to define a constructor with the static keyword a compile-time error will occur.

In general static class level. A constructor will be used to assign initial values for instance variables. Both static and constructors are different and opposite to each other. We need to assign initial values for an instance variable we can use a constructor. We need to assign static variables we can use static Block.

~~L~~ Constructor chaining is the process of calling one constructor from another constructor with respect to current object.

Constructor chaining can be done in two ways.

① Within same class: It can be done using this() keyword. For constructors in same class.

② From Base class: by using super() keyword to called constructor from the base class.

We need constructor chaining when we want to perform multitask in single constructor.

In Java when we create an object of the class it occupies some space in the memory. If we do not delete these object it remains in the memory and occupies unnecessary space that is not upright from the aspect of programming. To resolve this problem we use the destructor.

Destructor is the opposite of the constructor. The constructor is used initialize objects while the destructor is used to delete or destroy the object that realises the resource occupied by the object.

In java there is no concept about destructor but because of Thread which are automatically free up object garbage collection and memory. But there has a special method name finalizers that are non-deterministic.

In Java, the allocation and deallocation of objects handle by the garbage collector. The invocation of finalizers is not guaranteed because it invokes implicitly.

Advantages of Destructor:

- ① It ~~releases~~ releases the resource occupied by the object.
- ② No explicit call is required, it is automatically invoked at the end of the program execution.
- ③ It does not accept any parameter and cannot be overriden.

2 Static keyword can be used with class, variable, method and block. static members belong to the class instead of a specific instance, this means if you make a member static, you can access it without object. A static method can call without any object because when we make a member static it becomes class level.

Static members are common for all the instance object of the class but non static members are separate for each instance class.

Static block is used for initializing the static variables. This block gets executed when the class is loaded in the memory. A class can have multiple static blocks, which execute in the same sequence in which they have been written into the program.

O Static Block is a block of code with a static keyword

In general, these are used to initialize the static members. JVM executes static blocks before the main method at the time of class loading.

JVM first looks for the main method and then starts executing the program including static blocks. There are some protocols which governs the functioning of static blocks :-

- ① JRE first see the class loader
- ② Then execute all static things then static block
- ③ After that the task results in class loading is performed.

m There are few restrictions imposed on a static method

- ① The static method cannot use non-static data member or invoke non-static method directly
- ② The this and super cannot be used in static context.
- ③ The static method can access only static type data (Static type instance)
- ④ There is no need to create an object of the class to invoke the static method.
- ⑤ A static method cannot be overridden in a sub-class.

9

Scanner is a class which are found in `java.util` package. Java provides various ways to read input from keyboard, the `java.util.Scanner` class is one of them.

The Java Scanner Class breaks the input into tokens using a delimiter which is whitespace by default. It provides many methods to read and parse various primitive values. The Java Scanner class is widely used to parse text for strings and primitive types using a regular expression. It is the simplest way to get input in Java. The Java Scanner class extends `Object` class and implements `Iterator` and `Closable` interface.

(a)

"Answer to the Question no: 3"

JDK	JRE	JVM
Java Development Kit is a software development kit to develop applications in Java. In addition to JRE, JDK also contains number of development tools (compilers, Javadoc, Java Debugger etc.)	Java Run Environment is the implementation of JVM and is defined as a software package that provides Java class libraries along with Java Virtual Machine (JVM), and other components to run applications written in Java Programming.	Java Virtual Machine is an abstract machine that is platform-dependent and has three notions as a specification a document that describes requirement of JVM implementation, a computer program that meets JVM requirements, and instance an implementation that executes Java byte codes provides a runtime environment for executing Java byte code
JDK is primarily used for code execution and has prime functionality of development	On other hand JRE is majorly responsible for creating environment for code execution	
JDK is platform dependent for different platforms different platforms different JDK	Like of JDK JRE is also platform dependent	
JDK = Java Runtime Environment + Development tools	JRE = Java Virtual Machine + Libraries to run the Application	JVM on other hand specifies all the implementations and responsible to provide these implementations to JRE.
As JDK is responsible for prime development so it contains tool for developing, debugging and monitoring java application	On other hand JRE does not contain tools such as compiler or debugger etc. Rather it contains class libraries and other supporting files that JVM requires to run the program	JVM is platform independent
		JVM = Only Runtime environment for executing the Java byte code.

Implicit Type Casting

Implicit type Casting is performed by the compiler on its own when it encounters a mixed data type expression in the program's assistance. Implicit casting doesn't require a casting operator.

Example:-

```
int a = 42;
```

```
float b = a;
```

here b will contain typecast value of a, because while assigning value to b compiler typecasts the value of a into float then assign b.

Explicit Type Casterz

Explicit type casting is performed by the programmer. In this type casting programmer tells compiler to type cast one data type to another data type using type casting operator but there is some risk of information loss is there, so one needs to be careful while doing it.

Example:-

```
float a = 42.12;
```

```
int b = (int) a;
```

here we explicitly converted float value of a to int while assigning it to int b. (int) is the type casting operator with the type in which you want to convert

Autoboxing

Converting primitive value into an object of the corresponding wrapper class is called autoboxing.

Converting int to Integer class.

Passed as a parameter to a method that expects an object of the corresponding wrapper class.

Assigned to a variable of the corresponding wrapper class.

Example:

Integer i = new Integer(10)
Where Integer is wrapper class

Unboxing

Converting an object of wrapper to its corresponding primitive value is called unboxing.

Conversion of Integer to int.

Passed as a parameter to a method that expects a value of the corresponding primitive type.

Assigned to a variable of the corresponding primitive type

Example:

int i1 = i;

Where i1 is primitive data and assign value from wrapper class.

(d)

private	Default	Protected	Public
The access level of privi a private modifier is only within the classes. It cannot be accessed from outside the class.	The access level of a default modifier is only within the package. It cannot be accessed from the outside of package.	The access level of a protected modifier is within the package and outside the package through child class. If we don't make the child class, it cannot be accessed from outside the package.	The access level of a public modifier is everywhere. It can be accessed from within the class, within package and outside the package.

(e)

Constructor

Construction is used to initialize an object.

Constructors are invoked implicitly

Constructor does not return any value

Constructor is not present, a default constructor is provided by java compiler.

Constructor should be of the same name of class

Method

Method is used to exhibits functionality of an object.

Methods are invoked explicitly

Method may not return a value

Method has no default provider.

Method name can be different from class.

(f)

Stack

Static memory is used to store items which have a very short life local variables or a reference variable of objects.

The stack is always reserved in LIFO (Last In First Out) order.

We can increase stack memory size by using parameter -Xss

Variables are visible to only to owner thread

JVM will throw java.lang.StackOverflowError

Heap

Heap memory is allocated to store objects and JRE classes.

Heap memory is dynamic allocation there is no fixed pattern for allocating and deallocating blocks in memory

We can increase or decrease heap memory size by using JVM option -Xm and -Xmx

It is visible to all threads

JVM will throw java.lang.OutOfMemoryError

(8)

==

== is an operator

== should be used during reference comparison. == checks if both references points to same location or not.

== operator can ~~be~~ not be overridden.

equals() method

equals() is a method of Object class. equals() method should be used for content comparison. equals() method evaluates then content to check the equality.

equals() method if not present and Object.equals() method is utilized, otherwise it can be overridden.

(h)

Mutable

We can modify the states of a mutable object ~~as~~ after it is created.

Mutable objects are not thread safe.

Mutable classes are not final.

By default all classes and its objects are mutable by nature.

Immutable

We can't modify the state of the object as after it is created.

Immutable objects are thread safe.

To create an immutable object, make the class final.

String and all wrapper classes are the example of immutable classes.

(i)

String	StringBuilder	StringBuffer
String is a class of Java	StringBuilder also a class of Java	StringBuffer also a class of Java
String is immutable	StringBuilder is mutable	StringBuffer is mutable.
	StringBuilder method is not synchronized	StringBuffer methods are synchronized
String is thread-safe class	StringBuilder is not thread-safe class	StringBuffer is a thread-safe class
String is faster than other two classes	StringBuilder is faster than StringBuffer	StringBuffer is slower than StringBuilder
String is not time efficient	StringBuilder is time efficient	StringBuffer is time inefficient
String is not memory efficient	StringBuilder is memory efficient	StringBuffer is memory inefficient

(J)

static (class) method

Static methods in Java that can be called without creating an object of class.

Static keyword is declared with static keyword.

Static methods means which will exist as a single copy for a class.

Static methods can be invoked by using class reference.

Static methods can't access instance methods and instance variables directly.

instance method

Instance method are methods which require an object of its class to be created before it can be called.

Instance method is not with static keyword.

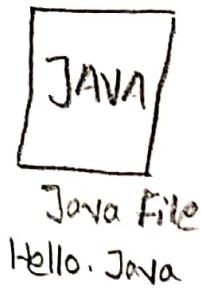
Instance methods exist as multiple copies depending on the number of instances created for that class.

Instance or non static methods are invoked by using object reference

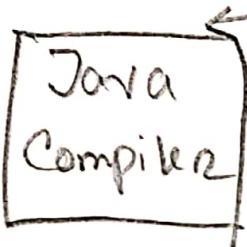
Instance method can access static variable and static methods directly.

• Answer to the Question no: 5 •

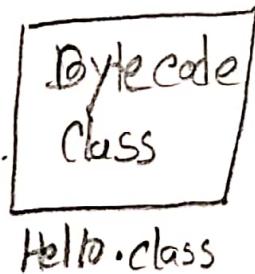
Figure: 1



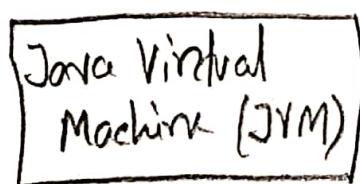
→ At first we save a file with .java name then we write the code into that file. after saving this file we call javac (Java Compiler)



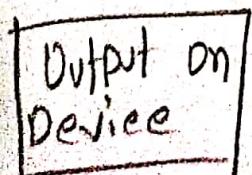
→ Java Compiler compile the full code into one by one line. After finish the compile if there have no error they convert the code to universal byte code. If they have any error it will be called compiler error.



→ Byte code class mean convert the full code to binary digit and make a .class file. By this Byte Code Class file we can run this code anywhere.



→ Java has a special Byte code decoder who again decode the Byte code class file and show the to the User screen. If there have any error it called run time error because the code is run by JVM.



→ After successfully run by JVM show the actual output to the device like Windows, Mac, Linux, Solar etc.

Figure: 2

Data Type: A data type or simply type is a attribute of data which tells the compiler how the programmers intends to use the data. There Are Two types of data in Java

① Primitive Data Type; A Basic type is a data type provided by a programming language as a basic building block. For example Java has Two type Primitive Data Type

① BooleanType: This data has conditional value like True & false which is define by boolean keyword.

② Alpha Numeric Type; This data has real number and alphabetic & sign or syntax.

① Integral Type: Who have full type of number on character data.

① Character Type: Only defines character on Alphabets or signs which keyword is char.

② Numeric Type: Only defines number which keyword is

① byte:

② short

③ int

④ long

⑪ Floating Point Type: This is also numeric data type but have decimal points which keywords is

① float

② double.

⑫ Non-primitive Data Types: These data types are not actually defined by programming language but are created by the programmer. They are also called "reference variable" or "object references" since they reference a memory location which store the data. There are two type of non-primitive data type

① Built in Library classes

② User Defined classes

① Built in Library classes: In java there are many type of build in library class like strings, Arrays, Classes, Interface

② User Defined Classes: Which are defined by ~~user~~ self.

Figure : 3

String is a non-primitive data so it's defined by class in Java there have a String class in java.util package. String can be declared by two type one is literal and another one is object reference, the main difference between literal and object string is memory allocation. For literal they make only one memory but for object they make each memory for each one. For example

```
String s1 = "java" → literals
String s2 = "java" → literals
String s3 = new String("java") → Object
```

When s1 and s2 are created s1 make a memory for their value java again when they see s2 also same value and same type that's why they don't create another memory gave the value to s1 that's things is called common-Pool. Another side s3 also set same value but created on declare as a object that's why they create a new memory.

Figure: 04

The static keyword in Java is used for memory management mainly. We can apply static keyword with variables, methods, blocks and nested classes. The static keyword belongs to the class than an instance of the class.

The static can be

- ① Variable (also known as a class variable)
- ② Method (also known as a class method)
- ③ Block
- ④ Nested class

① Java static variable:

If we declare any variable as static , it is known as a static variable.

The static variable can be used to refer to the common property of all objects.

The static variable gets memory only once in the class area at the time of class loading.

② Java static Method:

If we apply static keyword with any method, it is known as static method.

A static method belongs to the class neither the object of a class .

A static method can accessed static data member and can change the value it.

There are two main restrictions for the static method. They are:

- ① The static method can not use non-static data members or call non-static method directly.
 - ② this and super cannot be used in static context.
- ③ Static Block

It is used to initialize the static data member.

This is executed before the main method at the time of class loading.

Figure : 5

Following figure is about class and object.

A class is user defined blueprint or prototype. It represents the set of properties or methods that are common to all objects of one type. It is a basic unit of Object Oriented Programming and represent the real life entities. A typical Java program creates many objects, which as we know, interact by invoking methods. An object consist of

- (1) State
- (2) Behaviour
- (3) Identity.

When an object of class is created here Car is a class who can have some objects like name of car, color etc. the class is said to be instantiated. All the instances share the attributes and the behaviour of the class. A single class may have any number of instances. As we declare here like Car or Audi, Car Nissan, Car Volvo and this object can be many objects from that class. like color, model number, etc but they must ~~have~~ have the three comint.