Assignment 1

Q1 What do you know about JVM, JRE & JDK?

Ans: JVM :- Java virtual Machine

- JVM is responsible for interpreting the bytecode & translating it into machine code that can be executed by the underlying operating system.
- It is also provide memory management, security, & other runtime services necessary for executing java applications.

JRE :- Java virtual Machine

- JRE = rt.jar + java virtual Machine

JRE refers to a runtime environment in which java bytecode can be executed.

- JRE is a platform dependent software.
- To run java application on client's machine we must install JRE

JDK :- Java development kit

JDK = java development tools + java Docs + rt.jar + java virtual Machine

- It is the tool necessary to compile, document & package java program.
- It is platform dependent software.
- Developers must install JDK to direlope Java application.

Page No.
Date

(Q2) Is JRE platform dependent or independent?

Ans:- Yes, it is platfor dependent.
because;
to run java application on clients machine we must install JRE

(Q3) Which is ultimate base class in java class hierarchy? List the name of methods of it?

Ans:- Object class is ultimate base class in java class heirarchy.

Object class Method
1) toString () method
2) hashCode ()
3) equals (Object obj)
4) finalize ()
5) getClass ()
6) clone ()
7) wait (), notify (), notifyAll ()

(Q4) Which are the refirence types in java?

Ans:- Annotation
Array
class
Enumiration

Page No.
Date

Q5) Explain narrowing & widening ?

Ans: Narrowing :- process of converting value of variable of wider type into narrower type is called as narrowing.

eg.

```
double num1 = 10.5d;
int num2 = (int) num1; // Narrowing
```

Widening :- process of converting, value of variable of narrower type into wider type is called as widining.

eg.

```
int num1 = 10;
double num2 = (double) int1;
```

↖ widening

Q6) How will you print "Hello CDAC" statement on screen, without semicolon ? :

Ans using if statement

eg)

```
class Demo
{
public static void main (String args [])
  {
    if (System.out.println ("Hello CDAC") ==
        null) {
  }
```

Page No.
Date

Q.7) Can you write java application without main function ? if yes, how ?

Ans:- Yes, we can write java application without using main function. by using static block.

Static block in java (group of statement) that executed only once when the class is loaded into main memory by java. class loader it is also known as static block intilization.

eg)

```
public final class Demo {
    static
        {
            System.out.println("Hello");
        }
}
```

error:- Hello
java.lang. NoSuchMethodErro. Main

Q8) what will happen, if we call main method in static block ?

Ans Main method is entry point function in java program.
& Main method is typically called

by java virtual machine (JVM). It is possible to calling main method from static block. Main method should not be called manually in our program. Calling main method from static block may not setup program execution enivronment correctly & leading to unexpected result.

(Q9) In System.out.println, Explain meaning of every word?

Ans:- System.out.println → print passing argument & add new line after it.

print → printing argument
ln → new line
System.out → standard output stream

System refers to final class in java.lang.package

All the instance of the class printstream have a public method called println()

out is an instance of the printstream class.

(Q10) How will you pass object to the function by reference?

1) Making public number variable in class
2) Return value & update it

Ans → 3) create single element array

By this method, a single element array is created & passed as a parameter to the function; the effect is visible in the original memory address

```
class Main {
  public static void main (String [] args)
{
  int number [] = {1};
System.out.println ("number = " + number [0]);
  update (number);
System.out.println ("number = " + number [0]);
}
public static void update (int number []) {
   number [0] ++;
   }
}
```

(q11) Explain constructor chaining? How can we achieve it in C++?

Ans A constructor in java is used to initialize an instance of class. Also, a constructor can call other constructor of same class or super class, while constructor call must be very first call from other constructor & such a series of invocation of constructor

in c++, constructor do not have implicit constructor chaining like other language. However we can achieve it by explicitly calling one constructor from another in the same class.

Q12) which are rule to overload method in sub class ?

Ans 1) Same method name
2) Diffrent parameter (order)
3) same or wider access modifiers
4) diffent return type
5) throw same exception or their sub class

Q13) Explain the diffrince among finalize & dispose ?

Ans:	Finalize	Dispose
Defined	1) It's defined in interface IDisposable interface	It is defined in java.lang. object class
Basic →	It is used to close or release unmanged resources) held by object.	It is clear unmanged resource hold by current object before it destroyed.
Access specifier	public	private
Invoked	invoked by user	invoked by garbage collector
Implementation	Every time there is a close() function the dispose() method should be implemented	Unmanged (method) resource must be implemented using finalize() method

Page No.
Date

Q14) Explain diff. among final, finally & finalize?

Ans final :- final is keyword used in java to restrict the modification of a variable method or class.

finally :- finally is block used in java to ensure that a section of code is always executed, even if an exception is thrown.

finalize :- finalize is a method in java used to perform cleanup processing on an object before it is garbage collected

Q15) Explain diff among checked & unchecked exception?

Ans :- checked exception happen at complie time when the source code is transformed into executable code.

Uncheked exception happen at runtime when the executable program start running.

Q16) Explain exception chaining?

Ans :- Exception chaining is object oriented programming technique of handling exceptions. Exceptions chaining occurs when one one exception causes

exception. chained exception are associated such that previous exception cause each exception in the chain.

(q17) Explain the diff. among throw & throws ?

Ans :-

	throw	throws
point of usage	The throw keyword is used inside a function. It is used when it is required to throw exception logically	throws keyword is used in function signature. It is used when function has some statement that can lead to exception.
Exception thrown	Throw exception explicitly. one exception at a time	throws keyword declare multiple exception, seprated by comma.
Syntax	Syntax of throw keyword includes instance of exception.	Syntax of throws keyword includes the class name of exception to be thrown
propagation of exception	throw keyword cannot propogate checked exception	throws keyword is used to propogate the checked exception only.

Page No.
Date

Q.18) In which case, finally block doesn't execute?

Ans One of condition when the finally block may not get executed. If the jvm crashes or the program is terminated using system.exit() method.

In these case, the finally block will not execute

```
public static void main(String[] args)
{
  try {
      System.exit(0);
  }
  finally {
  System.out.println("this not executed");
  }
}
```

Q 19 Explain up casting?

Ans: Upcasting is the typecasting of a child object to a parent object. Upcasting can be done implicitly. Upcasting gives us the flexibility to acess the parent class mimbers but it is not possible to access all the child class member using this features

eg)
```
class Parent {
    void printData () {
    System.out.println ("method of parent class");
    }
}
class Child extends Parent {
    void Print Data () {
    System.out.println ("method of child class");
    }
}
class Upcasting Example {
    public static void main (String args[]) {
    parent obj1 = (Parent) new Child ();
    Parent obj2 = (parent) new Child ();
    obj1.printData ();
    obj2.printData ();
    }
}
```

Q20) Explain dynamic method dispatch?

Ans:- Dynamic method dispatch is the mechanism in which a call to an overriden method is resolved at runtime instead of compile time.

This is a runtime polymorphism.

Java uses the principle of "a superclass reference variable can refer to a subclass object" to resolve calls to overriden method at runtime. when a superclass reference is used to call an overridden method, java determine which version of the method to execute based on the type of object being referred to at the time call.

Advantage :-

1) It allows java to support overriding of methods, which are imp for runtime polymorphism
2) It allows a class to define method that will be shared by all its derived classes while also allowing these sub classes to define their specific implementation of a few or all of those method
3) It allow subclasses to incorporate their own method & define their implementation.

Q21) What do you know about final method?

Ans:- When a method is declared as final method in parent class, then any child class cannot override or modify the final method.

purpose of creating final method is to restrict the unwanted & improper use of method defination while overriding the parent class method.

Q22) Explain fragile base class problem & how can we overcome it?

Ans: The fragile base class problem refers to a situation in object oriented programming where modification to a base class can inadvertently cause issues or unexpected behaviour in derived classes

In other words, if we make changes in super class method then it is necessary to recompile all the subclasses is called fragile base class problem.

To overcome it :→ This problem could be avoided by declaring the method in the superclass as final which make impossible for subclass to

override them. However it is not always possible.
Therefore it is good practice for super classes to avoid changing calls to dynamically bound methods

(Q23) Why java does not support multiple implementation inheritance?

Ans: to prevent ambiguity, considir a case wher class B extends class A & class C & both class A & C have same method display(); now java complier cannot decide, which method it should inherit. To prevent such situation, multiple inheritances in not allowed in java.

(Q24) Explain marker interface? List the name of some marker interfaces?

Ans:- It is an empty interface (no field or method)

Name:-
Seiralizable
cloneble
Remote interface

A marker interface is an interface that docint have any methods or constant

It provides runtime type info. about object. so that compiler & jvm have additional information about the object It is also called as tagging interface.

(Q 25) Explain the significance of marker interface?

Ans Marker interface is used as a tag that inform the java compiler by a message so that it can add some special behaviour to the class implementing it.

It is usefull if we have info. about the class & info never chage in such case we use marker interface represent to represent the same. implementing an empty interface tell the compiler to do some operations.

It is used to logically divide the code & good way to recoganize code. It is more useful for developing API & in framework like spring.