IAT1 – JAVA and J2EE 10CS753
Model Answer/Solution
Dr. P. N. Singh

Q1 Briefly Explain JDK with JVM & JRE
Ans:

JDK(Java Development Kit) . It physically exists. It contains JRE + development tools.

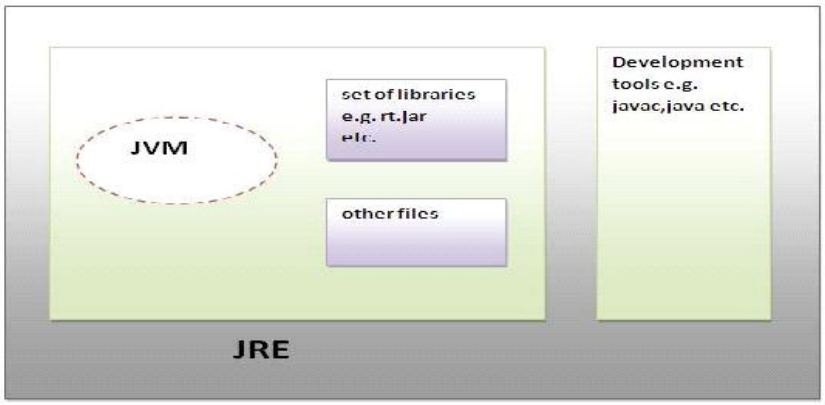
- javac(Java Compiler),
- java(Java Interpretor),
- Javap(Java diasssembler),
- Appletviewer,
- javah(for HTML docs)
- jdb(Java debugger)

JVM (Java Virtual Machine)
JVM is an abstract machine. It is a specification that provides runtime environment in which java bytecode can be executed.
JVMs are available for many hardware and software platforms. JVM, JRE and JDK are platform dependent because configuration of each OS differs. But, Java is platform independent.
The JVM performs following main tasks:

- Loads code
- Verifies code
- Executes code
- Provides runtime environment

JVM Notions: Specification & Implementation

JRE(Java Runtime Environment).
It is used to provide runtime environment. It is the implementation of JVM. It physically exists. It contains set of libraries + other files that JVM uses at runtime.
Implementation of JVMs are also actively released by other companies besides Sun Micro Systems.



JDK

Q2 (a) Correct Error in following code segment with explanation

```
byte b;
b=b*2;
```

Ans:

Error:

- First b is not initialized.
- Error:incompatible types :possible lossy conversion from int to byte.

Explanation:

When multiplying b(byte) with 2 (int) by java's automatic type promotion result will be an integer.But,'b'is of type byte .An integer value cannot be assigned to a byte variable without typecasting.

(b) Write the output of following code segment with explanation:

```
byte b;
int i = 200;
b = (byte) i;
System.out.println(b);
```

Ans:

Output: -128+200-128 = -56

Explanation: byte occupies 1 byte(8 bits) and integer occupies 4 bytes (32 bits) in java When it is signed and left most bit is 1 then it gives negative value:

Up to 127 it will store same value. If it is greater than 1then formula is : -128 + value -128 using 2's compliments: or directly : value -256 so, 200 – 256 will be -56

Q3 Write a java program to create multiplication table of number given by user during execution through keyboard

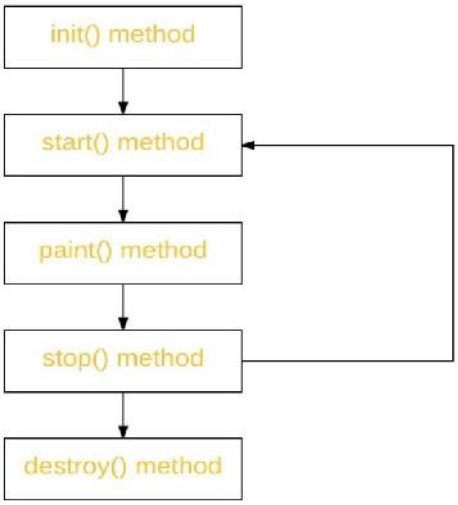
Ans:

```
//To create multiplication table of number entered through keyboard
import java.io.*;
class Multable{
  public static void main(String args[]) throws IOException {
  BufferedReader br = new BufferedReader(new
InputStreamReader(System.in));
  String str;
  int n,x;
  System.out.print("Enter number to create Multiplication Table: ");
  str=br.readLine();
  n=Integer.parseInt(str);
  System.out.println(" Multiplication Table of  " + n);
  for(x=1;x<=20;x++)
    System.out.println(n + " * " + x + " = " + n*x);
  }
}
Expected Output:
Enter number to create Multiplication Table: 5
Multiplication Table of 5
```

```
5 * 1 = 5
5 * 2 = 10
…..
5 * 20 = 100
```

Q4 Explain life cycle of an applet. Develop a Java applet (With HTML Tag) that sets background colour cyan, foreground colour red and outputs a string message "Welcome to CMRIT".

Solution:



Life Cycle of an applet

Lifecycle methods for Applet:

The java.applet.Applet class 4 life cycle methods and java.awt.Component class provides 1 life cycle methods for an applet.

java.applet.Applet class

For creating any applet java.applet.Applet class must be inherited. It provides 4 life cycle methods of applet.

public void init(): is used to initialized the Applet. It is invoked only once.

public void start(): is invoked after the init() method or browser is maximized. It is used to start the Applet.

public void stop(): is used to stop the Applet. It is invoked when Applet is stop or browser is minimized.

public void destroy(): is used to destroy the Applet. It is invoked only once.

java.awt.Component class

The Component class provides 1 life cycle method of applet.

public void paint(Graphics g): is used to paint the Applet. It provides Graphics class object that can be used for drawing oval, rectangle, arc etc.

```
//program
import java.applet.Applet;
import java.awt.*;
public class myapplet1 extends Applet{
     public void paint(Graphics g){
           setBackground(Color.cyan);
           setForeground(Color.red);
           g.drawString("Welcome to CMRIT",50,100);
     }
}
//Adding HTML tag to run by appletviewer
```

```
/*
 <applet code="myapplet1.class" width="300" height="300">
</applet>
*/
```

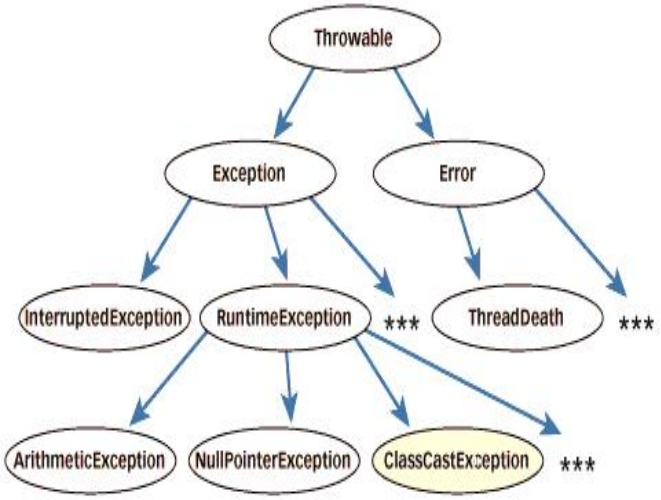
To run:
javac myapplet1.java
appletviewer myapplet1.java

Q5 What is Java exception ? Explain exception handling mechanism with try, catch and throw with example program.

Ans:

- "Exceptions are run-time errors" (sometimes compile time error also)
- Exception is an abnormal condition that arises in the code sequence occur during run time.(few at compile time)
- "throwable" is the super class in exception hierarchy.
- Compile time errors occurs due to incorrect syntax.
- Examples of run-time errors(exceptions)
  - Invalid/incorrect input by user
  - Divide by zero, square root of negative value
  - Wrong resource name(example: filename)
  - Subscript out of bounds
  - Numeric overflow
  - Logical error that was not fixed

- In Java, exceptions are also objects. Objects are thrown as exception whose classes descend from *throwable*.
- throwable serves as the base class for an entire family of exception classes, declared in java.lang utility.
- Exceptions are thrown to signal abnormal conditions that can often be handled by some catcher, though it's possible they may not be caught and therefore could result in a dead thread.
- Errors are usually thrown for more serious problems, such as OutOfMemoryError, that may not be so easy to handle. In general, code you write should throw only exceptions, not errors.
- Errors are usually thrown by the methods of the Java API, or by the Java virtual machine itself.



Java Exception Handling Keywords

There are 5 keywords used in java exception handling.

- try
- catch
- finally
- throw
- throws

```
//Solution by try-catch block
public class Testtrycatch2{
  public static void main(String args[]){
   try{
      int data=50/0;
   }catch(ArithmeticException e){System.out.println(e);}
   System.out.println("rest of the code...");
}
}
```

o/p:

Exception in thread main java.lang.ArithmeticException:/ by zero rest of the code...

```
class TestFinallyBlock1{
  public static void main(String args[]){
  try{
   int data=25/0;
   System.out.println(data);
  }
  catch(NullPointerException e){System.out.println(e);}
  finally{System.out.println("finally block is always executed");}
  System.out.println("rest of the code...");
  }
}
```

o/p:

task1 completed

rest of the code...

Q6. Explain final, finally and finalize of java.

Ans:

There are many differences between final, finally and finalize. A list of differences between final, finally and finalize are given below:

No.	Final	Finally	Finalize
1)	Final is used to apply restrictions on class, method and variable. Final class can't be inherited, final method can't be overridden and final variable value can't be changed	Finally is used to place important code, it will be executed whether exception is handled or not.	Finalize is used to perform clean up processing just before object is garbage collected.
2	Final is a keyword.	Finally is a block.	Finalize is a method.

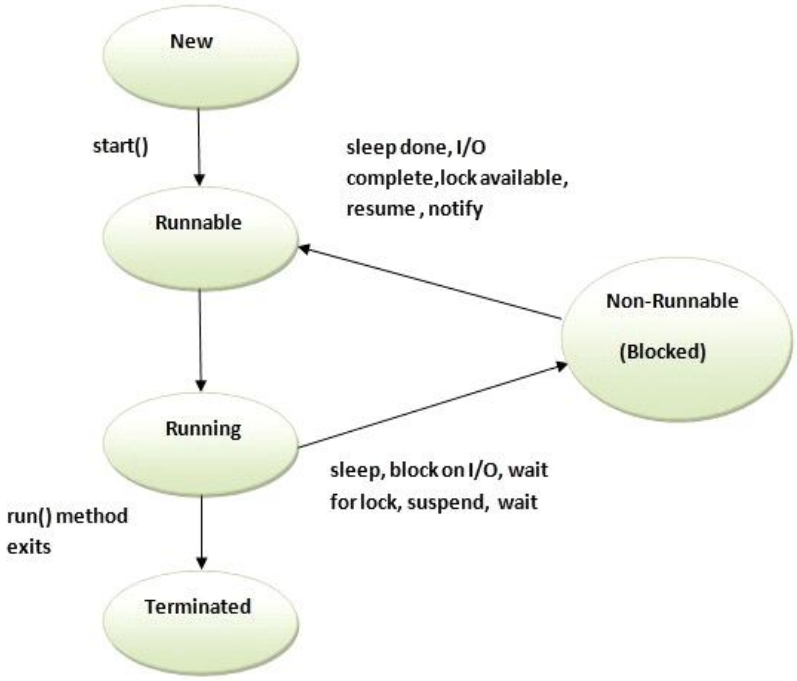
```
//Java final example
class FinalExample{
public static void main(String[] args){
 final int x=100;
x=200;//Compile Time Error
 }}

//Java finally example
 class FinallyExample{
public static void main(String[]args){
 try{
 int x=300;
 }
catch(Exception e){
System.out.println(e);
}
 finally{
System.out.println("finally block is executed");
} }}

//Java finalize example
 Class FinalizeExample{
public void finalize(){
System.out.println("finalize called");
}
 public static void main(String[] args){
 FinalizeExample f1=new FinalizeExample();
 FinalizeExample f2=new FinalizeExample();
 f1=null;
 f2=null;
 System.gc();
}}
```

Q7 (a) Describe life cycle of a thread with diagram ?

Ans:

A thread can be in one of the five states. According to sun, there is only 4 states in **thread life cycle in java** new, runnable, non-runnable and terminated. There is no running state.
But for better understanding the threads, we are explaining it in the 5 states.
The life cycle of the thread in java is controlled by JVM. The java thread states are as follows:

- New
- Runnable
- Running
- Non-Runnable (Blocked)
- Terminated



Life cycle of a Threaad

1) New : The thread is in new state if you create an instance of Thread class but before the invocation of start() method
2) Runnable : The thread is in runnable state after invocation of start() method, but the thread scheduler has not selected it to be the running thread.
3) Running: The thread is in running state if the thread scheduler has selected it.
4) Non-Runnable (Blocked) : This is the state when the thread is still alive, but is currently not eligible to run.
5) Terminated: A thread is in terminated or dead state when its run() method exits.

(b) Develop a Java program to find Factorial of a number using recursion.

Ans:

```
//To to find factorial of a number by recursion
import java.util.Scanner;
class Fact
{
    static int fact(int n)
   {
      if (n==1||n==0)
                  return 1;
      else
                  return (n*fact(n-1));
    }

   public static void main(String args[])
   {
       int num;
       Scanner s=new Scanner(System.in);
       System.out.println("Enter a number:");
       num=s.nextInt();
       //Fact f=new Fact();
       System.out.println("Factorial of"+num+" is "+fact(num));
   }
}
```