1.What is the difference between JDK and JRE?

Ans. The full form of JDK is Java Development Kit, while the full form of JRE is Java Runtime Environment, JDK is a software development kit whereas JRE is a software bundle that allows Java program to run. JDK is a superset of JRE, and contains everything that is in JRE, plus tools such as the compilers and debuggers necessary for developing applets and applications. JRE provides the libraries

2.What is Java Virtual Machine (JVM)?

Ans. A Java virtual machine is a virtual machine that enables a computer to run Java programs as well as programs written in other languages that are also compiled to Java bytecode. The JVM has two primary functions: to allow Java programs to run on any device or operating system (known as the "Write once, run anywhere" principle), and to manage and optimize

program memory.

3.What are the different types of memory areas allocated by JVM?

Ans. The memory in the JVM divided into 5 different parts:

Class(Method) Area.

Heap.

Stack.

Program Counter Register.

Native Method Stack.

4What is JIT compiler?

Ans. The JIT compiler helps improve the performance of Java programs by compiling bytecodes into native machine code at run time. The JIT compiler is enabled by default. When a method has been compiled, the JVM calls the compiled code of that method directly instead of interpreting it.

5.How Java platform is different from other platforms?

Ans. Java platform is a software-only platform that runs on the top of other hardware-based platforms, other platforms are mostly hardware software or hardware only and can be run only on hardware based. Programmer can develop Java code on any OS. Most of the other platforms do not have this capability.

6.Why people say that Java is 'write once and run anywhere' language?

Ans. Java is a platform independent language. In Java, the program is not converted to code directly understood by Hardware, rather it is converted to bytecode(.class file), which is interpreted by JVM, so once compiled it generates bytecode file, which can be run anywhere (any machine) which has JVM( Java Virtual Machine) and hence it gets the nature of Write Once and Run Anywhere.

7.What are primitive data types?

Ans. Primitive types are the most basic data types available within the Java language. There are : boolean , byte , char , short , int , long , float and double

8.Why it is not preferred to use float and double in financial applications, write a program to discribe the issue.

Ans. We use float and double to represent decimal numbers but the result of a floating-point number is not exact, which makes them unsuitable for any financial calculation which requires exact result and not an approximation.   
  
public class Test {

public static void main(String args[]) throws IOException {

//floating point calculation

double a1 = 2.15;

double a2 = 1.10;

System.out.println("difference between 2.15 and 1.0 using double is: " + (a1 - a2));

//Use BigDecimal for financial calculation

BigDecimal a3 = new BigDecimal("2.15");

BigDecimal a4 = new BigDecimal("1.10") ;

System.out.println("difference between 2.15 and 1.0 using BigDecimal is: " + (a3.subtract(a4)));

}

}

9. If a variable of primitive data type is not assigned, what does it contain?

Ans. unassigned instance variables printed default values. But unassigned local variable  raises compilation error.

10.Why do we suffix L with long, F with Float and D with double?

During compilation, all floating point numbers (numbers with decimal point) default to double. Therefore, if you don't want your number to double and just want it as float, you have to explicitly tell the compiler by adding a **f** or **F** at end of the literal constant, and D with double.

11.What happens when you assign a variable of primitive data type to another

variable of same type?

Ans. Casting between primitive types enables you to convert the value of one type to another primitive type.

12.What are reference data types?

Reference datatypes in java are those which contains reference/address of dynamically created objects

13.Write a program and put these below comments in code and explain the behaviour

Ans.

public class Test {

public static void main(String args[]) {

int i=2;

System.out.println(i);

// \u000A is a newline;

// look inside c:\users

}

}

// look inside c:\users

This will give error of illegal Unicode escape

Basically it's \n , nl = New line = \u000a in the ASCII table.