**ASSIGNMENT - 03/18/2021**

**Syllabus-Doubts and Explanation**

1 - What are the latest java version of in market?

The latest version of Java is Java 16 or JDK 16 released on March, 16th 2021 . JDK 17 is in progress with early-access builds and will become the next LTS (Long Term Support) JDK.

2 - When We install Java? What are basic folder we got?

Jdk 13.0.1 and jre 1.8.0\_25

3 - When java created? and by whom?

Java was originally developed by James Gosling at Sun Microsystems (which has since been acquired by Oracle) and released in 1995 as a core component of Sun Microsystems' Java platform.

4 - What are all application available in bin folder of Java Setup?

"Bin" folder is usually where the compiled files are copied to. For example, this folder will have the . class, . ... To describe this in another way, "src" folder contains the human readable code and "bin" contains the binary that the java virtual machine executes.

5 - What is WORA?

Java applications are called WORA (Write Once Run Anywhere). This means a programmer can develop Java code on one system and can expect it to run on any other Java-enabled system without any adjustment. This is all possible because of JVM.

6 - install java in your machine and write down all the issues which you faced during installation?

No Issues

7 - How to increase RAM associated to JVM in command line

which command increase RAM associated to JVM

Java command-line parameters to help control the RAM use of application:

Use -Xmx to specify the maximum heap size

Use -Xms to specify the initial Java heap size

Use -Xss to set the Java thread stack size

Use this syntax to specify the amount of memory the JVM should use:

-Xms64m or -Xms64M

-Xmx1g or -Xmx1G

8 - How to will you check JDK is installed in Machine?

Write Command to check JDK is installed in machine

Open the command prompt. Follow the menu path Start > Programs > Accessories > Command Prompt

Type: java -version and press Enter on your keyboard.

Result: A message similar to the following indicates that Java is installed and you are ready to use MITSIS via the Java Runtime Environment

Java version "1.4.x\_xx" (or a higher version)

Java<TM> SE Runtime Environment <build .....>

Java hotspot<TM> Client VM <build .....>

9 - Can we have multiple version of Java in single machine?

Can a user install multiple version of java in a single machine?

Yes.It is possible that you also need to change the PATH environment variable to use the correct java binary. ... We can install multiple versions of Java Development kits on the same machine using SDKMan.

**Java Setup**

1 - How to increase heap memory in eclipse?

1.Open Eclipse and in the toolbar menu, go to Run -> Run Configurations…

2.In the left pane of Run Configurations window, navigate to the Java Application node and select the Java application for which you need to increase the heap size. Then in the right pane, click on the Arguments tab.

In the VM Arguments section, type the following arguments:

-Xms[CUSTOM\_SIZE] – This means that your JVM will be started with Xms amount of memory

-Xmx[CUSTOM\_SIZE] – This means that your JVM will be able to use a maximum of Xmx amount of memory.

For example,

-Xms512M – JVM will start with an initial memory size of 512 MB

-Xmx1024M – and can use up to a maximum of 1024 MB (or 1 GB).

Click on Apply button to save the changes.

2 - Eclipse Command line arguments

Click on Run -> Run Configurations

Click on Arguments tab

In Program Arguments section , Enter your arguments.

Click Apply

3 - How to add plugins in eclipse?

visit Eclipse market place

1. Open Eclipse and in the toolbar menu, go to Help-> Eclipse Marketplace.
2. In Eclipse Marketplace window, navigate to recent or popular node and select the plugin you want to install
3. For Example, to install PyDev-Python IDE for Eclipse, just click on the Install button. Using the dialog box that comes up you can install the plug-in.

**ASSIGNMENT 03/19/2021**

**Java Architecture**

1 - What is basic difference between JDK, JRE and JVM?

JDK (Java Development Kit) enables developers to create Java programs that can be executed and run by the JRE (Java Runtime Environment) and JVM (Java Virtual Machine). The JRE is the part of Java that creates JVM. JVM is the Java platform component that executes source code

2 - Why java is platform independent?

When the Java program runs in a particular machine it is sent to java compiler, which converts this code into intermediate code called bytecode. ... JVM recognizes the platform it is on and converts the bytecodes into native machine code. Hence java is called platform independent language.

3 - Model Assignment

1 - Create a file with java extension and name is Test

2 - write a class Test in that file

3 - Inside that class write public static void main(String[] args)

4 - Inside main method, write System.out.println("Hello");

Code will look like this:

class Test{

public static void main(String[] args){

System.out.println("Hello");

}

}

5 - Compile this code

javac Test.java

6 - Execute this code

java Test

4 - How to run Java Command with JVM memory assignment?

use -xms and -xmx switches?

Use this syntax to specify the amount of memory the JVM should use:

-Xms64m or –Xms64M

-Xms1g or –Xms1G

5 - Can We run without name java file using cmd ?

No

6 - If we have one java file, whose name is not exact same like class present in it? Can java execute that?

Create a file with A.java

and create a class in that file with name as "B"

class B{

public static void main(String...args){

}

}

Check Compiler behavior

Compiles without an error

What if we made below change (make class public)

public class B{

public static void main(String...args){

}

}

Check Compiler behavior

Shows an error as :

Public type B must be defined in its own file

**Execute Java Class from CMD**

1 - Is Empty .java file name working?

No

create a java file with .java (should not have any name). and try to compile and execute it.

Note: Keep some 2-3 classes in that.

2 - Can we set classpath to another folder and execute our classes

Compile some java file. keep all class files in another folder and execute that.

set classpath in Java command

java -cp <<--classpath-->> classfilename

**ASSIGNMENT 03/22/2021**

**IDE Understanding**

1 - What is IDE? write 5 IDEs name for java?

IDE –Integrated Development Environment : An integrated development environment (**IDE**) is software for building applications that combines common developer tools into a single graphical user interface (GUI).

1. Eclipse
2. MyEclipse
3. IntelliJ IDEA
4. BlueJ
5. Kite

2 - How to increase Heap Space in eclipse.ini file?

How to increase heapsize in eclipse.ini file

The heap size is specified and set through an .ini file at Eclipse startup. That ini file is inside the ‘eclipse’ folder,

**-vmargs** which get passed to the Java Virtual Machine (VM):

* **-Xms**: initial start heap size
* **-Xmx**: maximum heap size

The -XX options are passed to the VM internally, and with newer VM’s they are ignored.

3 - What is Xmx and Xms variables?

The flag **Xmx** specifies the maximum memory allocation pool for a Java Virtual Machine (JVM), while **Xms** specifies the initial memory allocation pool. This means that your JVM will be started with **Xms** amount of memory and will be able to use a maximum of **Xmx** amount of memory.

4 - How to set different JRE in eclipse.ini file

-vm variable in eclipse.ini file.

eclipse.ini vm argument is useful when you have multiple JDK installation and you want to make sure that your eclipse runs on a specific JVM, rather than picking system configured jdk path. It must be defined before **-vmargs**.

**ShortCuts and Setup in Eclipse and IntellIJ**

1 - write 10 useful shortcut in eclipse

1. Code Assist (CTRL + Space)
2. Quick Fix (CTRL + 1)
3. Refactoring (ALT + SHIFT + T)
4. Source (ALT + SHIFT + S)
5. Surround With (ALT + SHIFT + Z)
6. Delete Rows (CTRL + D)
7. Call Hierarchy (CTRL + ALT + H)
8. Quick Type Hierarchy (CTRL + T)
9. Quick Outline (CTRL + O)
10. Show All Shortcuts (CTRL + SHIFT + L)

2 - How to change compiler version in Eclipse?

Step 1: Go to **Window -> Preferences**  
Step 2: Go to **Java -> Compiler**

Step 3: Select **Compiler compliance level**  
Step 4: To change default compliance settings, we need to uncheck **Use default compliance settings** and set the compiler java version for .class and source compatibility.

3 - is eclipse compiler better then java compiler?

Yes

4 - How to change jre version in eclipse?

Step 1: Go to **Window -> Preferences**  
Step 2: Go to **Java -> Installed JREs**  
Step 3: Click on **Add** and select **Standard VM** and then click on **Directory** and select JRE home or JDK home. We can add more than one JRE versions. Find the print screen

Step 4: Select the check box to activate the required version of JRE

**Obfuscation**

1 - What is obfuscation

**Obfuscation** is the obscuring of the intended meaning of communication by making the message difficult to understand, usually with confusing and ambiguous language

2 - Difference between Obfuscation and Encoding?

The purpose of **obfuscation** is to make something harder to understand, usually for the purposes of making it more difficult to attack or to copy. It, like **encoding**, can often be reversed by using the same technique that **obfuscated** it.

**Keywords**

1 - How many keywords available in java?

52 keywords

Write some 20 keywords available in java

1. **boolean** – A data type that can hold True and False values only
2. **break** – A control statement for breaking out of loops
3. **byte** – A data type that can hold 8-bit data values
4. **case** – Used in switch statements to mark blocks of text
5. **catch** – Catches exceptions generated by try statements
6. **char** – A data type that can hold unsigned 16-bit Unicode characters
7. **class** -Declares a new class
8. **continue** -Sends control back outside a loop
9. **default** -Specifies the default block of code in a switch statement
10. **do** -Starts a do-while loop
11. **double** – A data type that can hold 64-bit floating-point numbers
12. **else** – Indicates alternative branches in an if statement
13. **enum** – A Java keyword used to declare an enumerated type. Enumerations extend the base class.
14. **extends** -Indicates that a class is derived from another class or interface
15. **final** -Indicates that a variable holds a constant value or that a method will not be overridden
16. **finally** -Indicates a block of code in a try-catch structure that will always be executed
17. **float** -A data type that holds a 32-bit floating-point number
18. **for** -Used to start a for loop
19. **if** -Tests a true/false expression and branches accordingly
20. **implements** -Specifies that a class implements an interface
21. **import** -References other classes

2 - What is reserve Keyword?

Java reserved words are keywords that are reserved by Java functions or other uses that cannot be used as identifiers (e.g., variable names, function names, class names). If a reserved word was used as a variable, you would get an error or unexpected result.

3 - Why java consider goto as keyword?

as goto functionality is not available in java but still java consider goto as keyword.

Java does not support goto, it is reserved as a keyword just in case they wanted to add it to a later version.

* Unlike C/C++, Java does not have goto statement, but java supports **label**.
* The only place where a label is useful in Java is right before nested loop statements.
* We can specify label name with break to break out a specific outer loop.
* Similarly, label name can be specified with continue.

**ASSIGNMENT 03/25/2021**

**Number System**

1 - Covert below Decimal to Octal

20,100,56,78,1453,544

**package** Typeconversion;

**public** **class** Decimaltooctal {

**public** **static** **void** main(String[] args) {

// **TODO** Auto-generated method stub

**int** i=20;

**int** j=100;

**int** k=56;

**int** l=78;

**int** m=1453;

**int** n=544;

System.***out***.println(Integer.*toOctalString*(i));

System.***out***.println(Integer.*toOctalString*(j));

System.***out***.println(Integer.*toOctalString*(k));

System.***out***.println(Integer.*toOctalString*(l));

System.***out***.println(Integer.*toOctalString*(m));

System.***out***.println(Integer.*toOctalString*(n));

}

}

OUTPUT:

24

144

70

116

2655

1040

2 Covert below Decimal to Binary

20,39,49,28,126,156

**package** Typeconversion;

**public** **class** Decimaltobinary {

**public** **static** **void** main(String[] args) {

// **TODO** Auto-generated method stub

**int** i=20;

**int** j=39;

**int** a=49;

**int** b=28;

**int** c=126;

**int** d=156;

System.***out***.println((Integer.*toBinaryString*(i)));

System.***out***.println(Integer.*toBinaryString*(j));

System.***out***.println(Integer.*toBinaryString*(a));

System.***out***.println(Integer.*toBinaryString*(b));

System.***out***.println(Integer.*toBinaryString*(c));

System.***out***.println(Integer.*toBinaryString*(d));

}

}

OUTPUT:

10100

100111

110001

11100

1111110

10011100

3 - Covert below Decimal to Hexadecimal

245,45,67,98,712

**package** Typeconversion;

**public** **class** Decimaltohex {

**public** **static** **void** main(String[] args) {

// **TODO** Auto-generated method stub

**int** i=245;

**int** j=45;

**int** a=67;

**int** b=98;

**int** c=712;

System.***out***.println(Integer.*toHexString*(i));

System.***out***.println(Integer.*toHexString*(j));

System.***out***.println(Integer.*toHexString*(a));

System.***out***.println(Integer.*toHexString*(b));

System.***out***.println(Integer.*toHexString*(c));

}

}

OUTPUT:

f5

2d

43

62

2c8

**Data Types in Java**

1 - Write Size and range for all data types?

DATATYPE SIZE RANGE

Byte- 1 byte- -128 to 127

Short- 2 bytes- -32,768 to 32,767

Int- 4 bytes- -2,147,483,648 to 2,147,483, 647

Long- 8 bytes -9,223,372,036,854,775,808 to  
 9,223,372,036,854,775,807

Float- 4 bytes - approximately ±3.40282347E+38F  
 (6-7 significant decimal digits)

Double- 8 bytes- approximately ±1.79769313486231570E+308  
 (15 significant decimal digits)

Char - 2 bytes- 0 to 65,536 (unsigned)

Boolean- true or false

2 - is below statement valid in java?

int int = 10;

if yes, why?else if no, why?

Not valid

Syntax error, insert "VariableDeclarators" to complete LocalVariableDeclaration

Syntax error, insert ";" to complete BlockStatements

**Numerical Data Type and Cyclic Nature**

1 - What is type casting?

When the data is converted from one data type to another data type, then it is called type casting. Type casting is nothing but changing the type of the data. Using type casting, only type of the data can be changed but not the data itself.

2 - int ln = 1234\_2343\_22376;

is this valid?

3 - solve below code

byte b1 = 128;

what is the output?

Syntax error:

Type mismatch: cannot convert from int to byte

byte b1 = (byte)b1;

what is the output?

Syntax error:

Type mismatch: cannot convert from int to byte

Duplicate local variable b1

byte b1 = (byte)(b1\*3);

what is the output?

Type mismatch: cannot convert from int to byte

Duplicate local variable b1

4 - solve below code

byte b1 = (byte)250;

byte b1 = (byte)120;

byte b1 = (byte)130;

byte b1 = (byte)300;

write output of b1;

5 - solve below code

int i = 020;

int j = 20;

int k = 0X20;

print output?

**package** Priya;

**public** **class** assignment {

**public** **static** **void** main(String[] args) {

// **TODO** Auto-generated method stub

//byte b1=128;

//byte b1=(byte)250;

//byte b1=(byte)120;

//byte b1=(byte)130;

//byte b1=(byte)300;//

**int** i=020;

**int** j=20;

**int** k=0X20;

//System.out.println(b1);

System.***out***.println(i);

System.***out***.println(j);

System.***out***.println(k);

}

}

OUTPUT:

16

20

32