Assignment # 01

Name	:-	Ahmad Baseer
Section	:-	C
Roll no	:-	124
Course	:-	OOP
Program	:-	B.S(CE)

Q1. List down most popular object oriented languages. Describe important features of Java Language.

Ans. As we know there are many programming languages but after concluding my study i found five Most popular oop languages which are:

- **Java**
- **Python**
- **Ruby**
- **Golang**
- **C++**

Important Features of Java:

Of course sure there may be many important features of java but yet the features i know i will try to list them:

- It can be used in small hardware resources like set-top boxes and handled devices.
- As java is statically typed language (Means its code will be check at compile time) so it gives faster execution.
- It is best for software engineer roles and it can handle system for large corporations.
- In last as java is a oop language so it has also the features of oop language which are Abstraction, Encapsulation, Inheritance, Polymorphism.

Q2. Define Math Class and which associated functions can be used for the following task with proper function syntax and description should be also given.

- **To get absolute value**
- **To get ceil of number**
- **To get floor of number**
- **To get max of two values**
- **To get min of two values**
- **To generate random number**
- **To calculate power for a number**
- **To calculate square root of number**

Ans. Definition:

It is a class in java which provides several functions or we can say methods to perform mathematical operations. It is located in java.lang package so it is built-in imported that's why we don't have to import that class.

- **To get absolute value:**

Source code:

```
package mathfunctions;
public class MathFunctions {
public static void main(String[] args) {
System.out.println("Absolute value of -3 is:"+Math.abs(-3));
}
   }
```

Description:

First we describe the package name to show what we are going to use and then we set up a class with security type "public" and with name which shows the purpose of our code. In second last step we set up a main method again with security type "public", and with keyword "static" which means it is not necessary to create its object for using this method. Finally, we setup a method to print the particular thing, we wrote that text to prompt user which task we did, text is written in double quotes b/c we want to print as it is then "+" is used to join the string and answer. We use "Math.abs()" method to calculate absolute value where "Math" is a class and "abs" is method which is defined in it so to access that method from Math class we use "dot". So finally with this method we can calculate absolute value.

- **To get ceil of number:**

Source code:

```
package mathfunctions;
public class MathFunctions {
public static void main(String[] args) {
System.out.println("Ceiling value of 3.25 is:"+Math.ceil(3.25));
System.out.println("Ceiling value of 3.96 is:"+Math.ceil(3.96));
}
   }
```

Description:

First we describe the package name to show what we are going to use and then we set up a class with security type "public" and with name which shows the purpose of our code. In second last step we set up a main method again with security type "public", and with keyword "static" which means it is not necessary to create its object for using this method. Finally, we setup a method to print the particular thing, we wrote that text to prompt user which task we did, text is written in double quotes b/c we want to print as it is then "+" is used to join the string and answer. We use "Math.ceil()" method to calculate ceiling value which means it will round of and give the nearest highest whole number no matter what the number is, so for describing we take two numbers as we can see "3.25" is nearly "3" but it will give "4" b/c 3 is not the nearest highest whole number, where "Math" is a class and "ceil" is method which is defined in it so to access that method from Math class we use "dot". So finally with this method we can calculate ceiling value.

- **To get floor of number:**

Source code:

```
package mathfunctions;
public class MathFunctions {
public static void main(String[] args) {
System.out.println("Floor value of 3.25 is:"+Math.floor(3.25));
System.out.println("Floor value of 3.96 is:"+Math.floor(3.96));
}
   }
```

Description:

First we describe the package name to show what we are going to use and then we set up a class with security type "public" and with name which shows the purpose of our code. In second last step we set up a main method again with security type "public", and with keyword "static" which means it is not necessary to create its object for using this method. Finally, we setup a method to print the particular thing, we wrote that text to prompt user which task we did, text is written in double quotes b/c we want to print as it is then "+" is used to join the string and answer. We use "Math.floor()" method to calculate ceiling value which means it will round of and give the nearest lowest whole number no matter what the number is, so for describing we take two numbers as we can see "3.96" is nearly "4" but it will give "3" b/c 4 is not the nearest lowest whole number, where "Math" is a class and "floor" is method which is defined in it so to access that method from Math class we use "dot". So finally with this method we can calculate floor value.

- **To get max of two values:**

Source code:

```
package mathfunctions;
public class MathFunctions {
public static void main(String[] args) {
System.out.println("Max number from the two is:"+Math.max(6,9));
System.out.println("Max number from the two is:"+Math.max(2,3)); }  }
```

Description:

First we describe the package name to show what we are going to use and then we set up a class with security type "public" and with name which shows the purpose of our code. In second last step we set up a main method again with security type "public", and with keyword "static" which means it is not necessary to create its object for using this method. Finally, we setup a method to print the particular thing, we wrote that text to prompt user which task we did, text is written in double quotes b/c we want to print as it is then "+" is used to join the string and answer. We use "Math.max()" method to calculate max value where "Math" is a class and "max" is method which takes two values or we can say parameters and returns the maximum value it is defined in it so to access that method from Math class we use "dot". So finally with this method we can calculate max value.

- **To get min of two values:**

Source code:

```
package mathfunctions;
public class MathFunctions {
public static void main(String[] args) {
System.out.println("Max number from the two is:"+Math.min(4,5));
System.out.println("Max number from the two is:"+Math.min(1,9));
}
    }
```

Description:

First we describe the package name to show what we are going to use and then we set up a class with security type "public" and with name which shows the purpose of our code. In second last step we set up a main method again with security type "public", and with keyword "static" which means it is not necessary to create its object for using this method. Finally, we setup a method to print the particular thing, we wrote that text to prompt user which task we did, text is written in double quotes b/c we want to print as it is then "+" is used to join the string and answer. We use "Math.min()" method to calculate max value where "Math" is a class and "min" is method which takes two values or we can say parameters and returns the minimum value it is defined in it so to access that method from Math class we use "dot". So finally with this method we can calculate min value.

- **To generate random number:**

Source code:

```
package randomgenerator;
import java.util.Random;
public class Randomgenerator {
public static void main(String[] args) {
Random RNum =new Random();
int Random_num=0+RNum.nextInt();
System.out.println("Your random number is:"+Random_num);
}
   }
```

Description:

First we describe the package name to show what we are going to use and then import java.util package to access the "Random" class we access it by placing "dot" in b/w it and package. Then set up a class with security type "public" and with name which shows the purpose of our code. In fourth step we set up a main method again with security type "public", and with keyword "static" which means it is not necessary to create its object for using this method. The next step is to create object of random class b/c class is nothing without object, we use "new" which will reserve memory for object and then we use constructor which will construct the object then we use this method "0+RNum.nextInt()" to generate random number and for holding the generated number i stored it in a variable.
Finally, we setup a method to print the particular thing, we wrote that text to prompt user which task we did, text is written in double quotes b/c we want to print as it is then "+" is used to join the string and answer.

- **To calculate power for a number:**

Source code:

```
package mathfunctions;
public class MathFunctions {
public static void main(String[] args) {
System.out.println("3 raised to power 2 is:"+Math.pow(3,2)); }  }
```

Description:

First we describe the package name to show what we are going to use and then we set up a class with security type "public" and with name which shows the purpose of our code. In second last step we set up a main method again with security type "public", and with keyword "static" which means it is not necessary to create its object for using this method. Finally, we setup a method to print the particular thing, we wrote that text to prompt user which task we did, text is written in double quotes b/c we want to print as it is then "+" is used to join the string and answer. We use "Math.pow()" method to calculate power of a no: where "Math" is a class and "pow" is method which takes two values or we can say parameters, second parameter will be power of first one it is defined in it so to access that method from Math class we use "dot". So finally with this method we can calculate power.

- **To calculate square root of number:**

Source code:

```
package mathfunctions;
public class MathFunctions {
public static void main(String[] args) {
System.out.println("Square root of 100 is:"+Math.sqrt(100));
}
  }
```

Description:

First we describe the package name to show what we are going to use and then we set up a class with security type "public" and with name which shows the purpose of our code. In second last step we set up a main method again with security type "public", and with keyword "static" which means it is not necessary to create its object for using this method. Finally, we setup a method to print the particular thing, we wrote that text to prompt user which task we did, text is written in double quotes b/c we want to print as it is then "+" is used to join the string and answer. We use "Math.sqrt()" method to calculate square root of a no: where "Math" is a class and "sqrt" is method it is defined in it so to access that method from Math class we use "dot". So finally with this method we can calculate square root.

Q3. List the different IDE in Java. Which IDE will you select for implementation of programs and why? Justify your answer.

Ans. As there are many IDE for java but some of them are listed below:

1. NetBeans
2. Eclipse
3. IntelliJ IDEA
4. Blue J
5. DrJava

Which IDE is my favorite and why:

Ans. Net beans is my favorite that's why I put it on first. So, why it is mine favorite? To justify my selection i will compare it with other two most popular IDE's which will automatically justify my selection the IDE's are: Eclipse and IntelliJ IDEA.

- If we talk about Intellij IDEA it has two versions which are:

1. Professional(Ultimate) version.
2. Community Version.

So the first one version is not free it means we have to pay for using its features while second one is free for use but it has lack of features while at the same time NetBeans is free for use and it has also those and much more features which are totally free so why we choose this one.

- If we talk about Eclipse so we have to download plugins means it can support them but it has not Built in those feature like html, javafx and javascript. If we don't download external plugins so it can't Support these features so why we download these as we already download this heavy software so What is best in it if it can't provide these features built in so these are some reasons due to NetBeans Is my favorite.

Q4. List the importance of type casting. Using your roll number write syntax of type casting into other data types.

Ans. Source code:

```
package typecasting;
public class TypeCasting {
public static void main(String[] args) {
// Widening type casting
byte a= 124;
short b=a;
int c=b;
long d=c;
float e=d;
double f=e;
```

```
// Narrowing type casting
double g= 124;
float h=(float)g;
long i=(long)h;
int j=(int)i;
char k=(char)j;
short l=(short)k;
byte m=(byte)l;
System.out.println(f);
System.out.println(k);
```

Importance of type casting:

To show the importance let's talk with some examples. Suppose, you have a task to to divide two numbers and prints the quotient and you are working in that manner:

```
byte a=4;
byte b=5;
float c=a/b;
```

Here you thought that 5 and 4 are not so big so why we use big (According to memory) data types to store them and then you use float variable to store the result of their division as you know the answer will be real value. The answer should be "0.8" as expected but after execution you got "0.0" why? b/c you are performing division on two integers so the answer will be in integer no matter in which will be variable you are storing the answer. So the integer answer of "0.8" is "0" but compiler will print "0.0" b/c you are storing it in float variable so it will print as a real value. If you want the answer which is expected So you have to type cast one of the operand into float and as we know there are many situations in which compiler automatically type cast the variable according to situation but these types of problems happens when compiler don't do it automatically as we see that it's a so simple program so just think what will happen if the concept of type casting does not exist, that is the importance of type casting which i try to explain.

Q5. Construct a program that prompts the user to input a positive integer. It should then print the multiplication table of that number.

Ans. Source code:

```
package table;
import java.util.Scanner;
public class Table{
public static void main(String[] args){
int no, i=1;
Scanner sc= new Scanner(System.in);
System.out.print("Please Enter any No:");
no= sc.nextInt();
while(i<=10){
System.out.printf("%d *  %d= %d\n", no , i , (no+no*(i-1)));
i++;  }    }   }
```

Output:

```
Please Enter any No:5
5 *  1= 5
5 *  2= 10
5 *  3= 15
5 *  4= 20
5 *  5= 25
5 *  6= 30
5 *  7= 35
5 *  8= 40
5 *  9= 45
5 *  10= 50
```

Q6. Construct the following into Java program called SumMinMax.java. Take exactly four integers and determines and prints the smallest, the largest and the sum of these four integers. The program's output should look like the following:

Sample Output:

The first integer: 4
The second integer: 6
The third integer: 2
The fourth integer: 7
The smallest integer was: 2
The largest integer was: 7
The sum of these four integers is: 19

Ans. Source code:

```
package summinmax;
import java.util.Scanner;
public class SumMinMax{
public static void main(String []args){
Scanner sc=new Scanner(System.in);
System.out.println("The first integer is:");
int a= sc.nextInt();
System.out.println("The Second integer is:");
int b= sc.nextInt();
System.out.println("The Third integer is:");
int c= sc.nextInt();
System.out.println("The fourth integer is:");
int d= sc.nextInt();
if(a<b && a<c && a<d){
System.out.print("The Smallest integer was:"+a);
}
if(b<a && b<c && b<d){
System.out.print("The Smallest integer was:"+b);
}
if(c<a && c<b && c<d){
System.out.print("The Smallest integer was:"+c);
}
if(d<a && d<b && d<c){
System.out.print("The Smallest integer was:"+d);
}
if(a>b && a>c && a>d){
System.out.print("The Largest integer was:"+a);
}
if(b>a && b>c && b>d){
System.out.print("The Largest integer was:"+b);
}
if(c>a && c>b && c>d){
System.out.print("The Largest integer was:"+c);
}
if(d>a && d>b && d>c){
System.out.print("The Largest integer was:"+d);
}
System.out.print("The Sum of these integers is:"+(a+b+c+d));  }   }
```

Output:

```
The first integer is:4
The Second integer is:6
The Third integer is:2
The fourth integer is:7
The Smallest integer was:2
The Largest integer was:7
The Sum of these integers is:19
```