

## INTERNAL ASSIGNMENT

**Course Code: OMC209**

**Last Date of Submission: 31/07/24**

**Course Title: Advanced Java Programming Laboratory**

**Maximum Marks: 30**

Program 1:

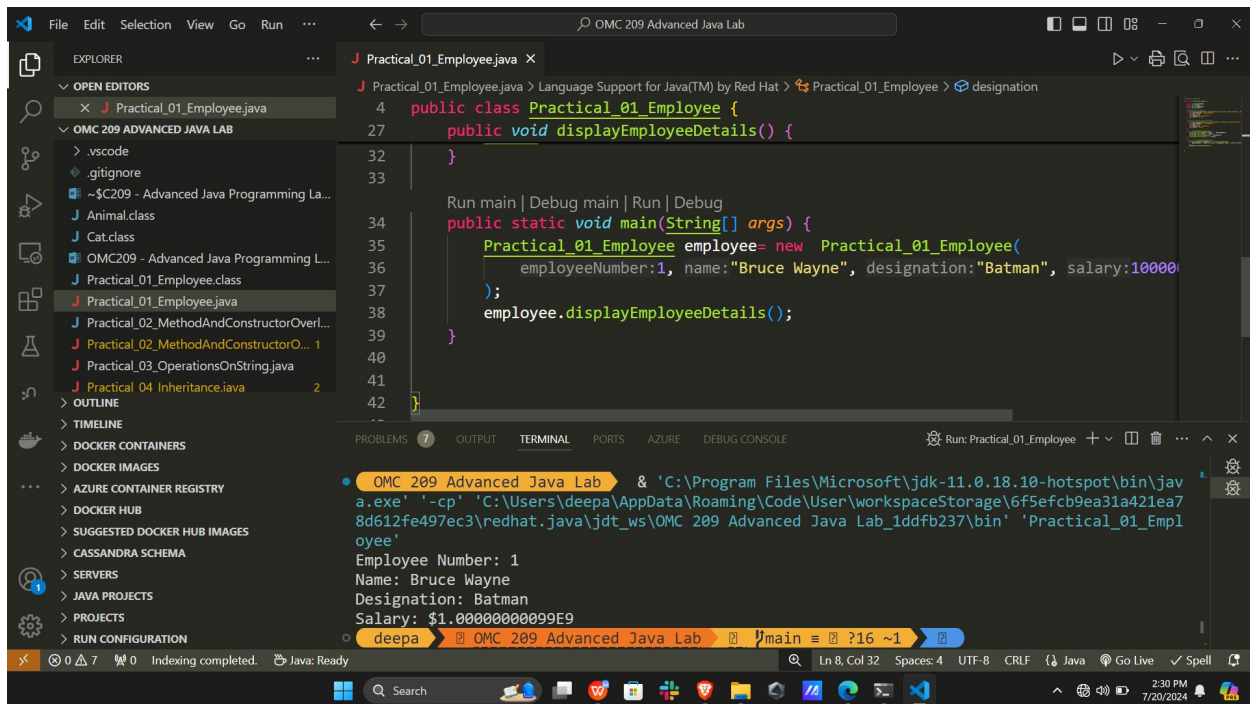
Create a Java class "Employee" with the attributes - employee number, name, designation and salary.

a. Implement methods to set and display these attributes

b. Implement parameterized constructors for initializing these attributes.

Use "this" keyword to illustrate the difference between instance variable and constructor parameters. Display these attributes.

Output:



```
File Edit Selection View Go Run ...
J Practical_01_Employee.java x
J Practical_01_Employee.java > Language Support for Java(TM) by Red Hat > Practical_01_Employee > designation
4 public class Practical_01_Employee {
27 public void displayEmployeeDetails() {
32 }
33
34 Run main | Debug main | Run | Debug
35 public static void main(String[] args) {
36     Practical_01_Employee employee= new Practical_01_Employee(
37         employeeNumber:1, name:"Bruce Wayne", designation:"Batman", salary:10000
38     );
39     employee.displayEmployeeDetails();
40 }
41
42
PROBLEMS 7 OUTPUT TERMINAL PORTS AZURE DEBUG CONSOLE
Run: Practical_01_Employee
• OMC 209 Advanced Java Lab & 'C:\Program Files\Microsoft\jdk-11.0.18.10-hotspot\bin\jav
a.exe' '-cp' 'C:\Users\deepa\AppData\Roaming\Code\User\workspaceStorage\6f5efcb9ea31a421ea7
8d612fe497ec3\redhat.java\jdt_ws\OMC 209 Advanced Java Lab_1ddfb237\bin' 'Practical_01_Emp
loyee'
Employee Number: 1
Name: Bruce Wayne
Designation: Batman
Salary: $1.000000000099E9
deepa OMC 209 Advanced Java Lab main ?16 ~1
Ln 8, Col 32 Spaces: 4 UTF-8 CRLF Java Go Live Spell
2:30 PM 7/20/2024
```



# INTERNAL ASSIGNMENT



**Graphic Era**  
Deemed to be University

```
1  .cal_03_OperationsOnString {
16 void palindromeCheck(String s){
17     Kstr= s.toLowerCase();
18     rse_s= new StringBuilder(checkstr).reverse().toString();
19     r.equals(reverse_s){
20         out.println(String.format(format:"'%s' is a palindrome.", s));
21     m.out.println(String.format(format:"'%s' isn't a palindrome.", s));
22
23
24 Run main | Debug main | Run | Debug
25 void main(String[] args) {
26     }3_OperationsOnString.palindromeCheck(s:"The Amazing Spiderman 3");
27     }3_OperationsOnString.charDigitCount(s:"The Amazing Spiderman 3");
28
29
30 OMC 209 Advanced Java Lab & 'C:\Program Files\Microsoft\jdk-11.0.18-hotspot\bin\jav
31 a.exe' '-cp' 'C:\Users\deepa\AppData\Roaming\Code\User\workspaceStorage\6f5efcb9ea31a421ea7
32 8d612fe497ec3\redhat.java\jdt_ws\OMC 209 Advanced Java Lab_1ddfb237\bin' 'Practical_03_Oper
33 ationsOnString'
34 ''The Amazing Spiderman 3'' isn't a palindrome.
35 Length of the given string: 23
36 Total number of non-digit characters in the string are: 22
37 Total number of digit characters in the string are: 1
38
39 deepa OMC 209 Advanced Java Lab main ?16 ~1
```

Program 4:

Write a Java program to demonstrate single and multi-level inheritance. Display the order of execution of constructors in multi-level inheritance.

Output:

# INTERNAL ASSIGNMENT



**Graphic Era**  
Deemed to be University

The screenshot shows an IDE with the following content:

```
18 class LevelTwoChild extends LevelOneChild{
19     public LevelTwoChild() {
20         System.out.println(x:"Level 2 child Constructor.");
21     }
22 }
23
24 }
25
26 public class Practical_04_Inheritance {
27     Run main | Debug main | Run | Debug
28     public static void main(String[] args) {
29         LevelTwoChild obj= new LevelTwoChild();
30     }
31 }
```

Terminal Output:

```
OMC 209 Advanced Java Lab & 'C:\Program Files\Microsoft\jdk-11.0.18-hotspot\bin\jav
a.exe' '-cp' 'C:\Users\deepa\AppData\Roaming\Code\User\workspaceStorage\6f5efcb9ea31a421ea7
8d612fe497ec3\redhat.java\jdt_ws\OMC 209 Advanced Java Lab_1ddfb237\bin' 'Practical_04_Inhe
ritance'
Base Class Constructor.
Level 1 child Constructor.
Level 2 child Constructor.
```

Program 5:

Write a Java program to demonstrate method overriding.

Output:

# INTERNAL ASSIGNMENT



**Graphic Era**  
Deemed to be University

```
8 class Dog extends Animal{
13 }
14 }
15
16
17 public class Practical_05_MethodOverriding {
18     Run main | Debug main | Run | Debug
19     public static void main(String[] args) {
20         Animal animal= new Animal();
21         Dog dog= new Dog();
22         animal.speak();
23         dog.speak();
24     }
25 }
```

PROBLEMS 12 OUTPUT TERMINAL PORTS AZURE DEBUG CONSOLE

• OMC 209 Advanced Java Lab & 'C:\Program Files\Microsoft\jdk-11.0.18-hotspot\bin\java.exe' '-cp' 'C:\Users\deepa\AppData\Roaming\Code\User\workspaceStorage\6f5efcb9ea31a421ea78d612fe497ec3\redhat.java\jdt\_ws\OMC 209 Advanced Java Lab\_1ddfb237\bin' 'Practical\_05\_MethodOverriding'

The animal makes a sound.  
The dog barks.

deepa OMC 209 Advanced Java Lab main 16 ~1

Program 6:

Write a Java program to illustrate the use of

- Abstract class
- Interfaces in Java

Output:



# INTERNAL ASSIGNMENT



**Graphic Era**  
Deemed to be University

```
File Edit Selection View Go Run ...
OMC 209 Advanced Java Lab

EXPLORER
  OPEN EDITORS
    Practical_01_Employee.java
    Practical_02_MethodAndConstru... 1
    Practical_03_OperationsOnString.java
    Practical_04_Inheritance.java 2
    Practical_05_MethodOverriding.j... 1
    Practical_06_AbstractClassesAnd... 2
  OMC 209 ADVANCED JAVA LAB
    Practical_02_MethodAndConstructorOverl...
    Practical_02_MethodAndConstructorO... 1
    Practical_03_OperationsOnString.java
    Practical_04_Inheritance.java 2
    Practical_05_MethodOverriding.java 1
    Practical_06_AbstractClassesAndInterf...
    Practical_06_AbstractClassesAndInterf... 2
  OUTLINE
  TIMELINE
  DOCKER CONTAINERS
  DOCKER IMAGES
  AZURE CONTAINER REGISTRY
  DOCKER HUB
  SUGGESTED DOCKER HUB IMAGES
  CASSANDRA SCHEMA
  SERVERS
  JAVA PROJECTS
  PROJECTS
  RUN CONFIGURATION

PRACTICAL_06_AbstractClassesAndInterfaces.java
55 public class Practical_06_AbstractClassesAndInterfaces {
56     Run main | Debug main | Run | Debug
57     public static void main(String[] args) {
58         Snake snake= new Snake(name:"Kaala Naag");
59         Cat cat= new Cat(name:"Persian Tommy");
60         cat.move();
61         snake.crawl();
62         snake.talk();
63         cat.talk();
64     }
65 }

TERMINAL
Java Lab\ " ; if ($?) { javac Practical_06_AbstractClassesAndInterfaces.java } ; if ($?) {
java Practical_06_AbstractClassesAndInterfaces }
This is a Kaala Naag, a Reptile.
This is a Persian Tommy, an Animal.
Persian Tommy moves.
Kaala Naag crawls.
Kaala Naag hisses.
Persian Tommy meows.

deepa OMC 209 Advanced Java Lab main ?16 ~1
```

## Program 7:

Write a Java program to demonstrate exception handling. Show the order of execution of "try", "catch" and "finally" blocks when an exception occurs and when it does not occur during the execution by providing appropriate inputs during execution and displaying messages.

Output:

# INTERNAL ASSIGNMENT



**Graphic Era**  
Deemed to be University

```
1 public class Practical_07_ExceptionHandling {
2     public static void main(String[] args) {
3         try {
4             int[] arr = new int[5];
5             arr[5] = 30 / 0;
6         } catch (ArithmeticException e) {
7             System.out.println(String.format("Exception occurred: [[ %s ]]", e));
8             System.out.println(x:"Arithmetic Exception' caught.");
9         } catch (ArrayIndexOutOfBoundsException e) {
10            System.out.println(String.format("Exception occurred: [[ %s ]]", e));
11            System.out.println(x:"Array Index Out Of Bounds Exception' caught.");
12        } finally {
13            System.out.println(x:"finally' block executed.");
14        }
15    }
16 }
```

Exception occurred: [[ java.lang.ArithmeticException: / by zero ]]  
'Arithmetic Exception' caught.  
'finally' block executed.

Program 8:

Write a Java program to illustrate the use of I/O streams.

Program 9:

Write a Java Servlet program to implement a dynamic HTML using Servlet( Student name and enrolment number should be accepted using HTML and displayed using a Servlet).

Program 10:

Write a Java Servlet program to demonstrate the use of cookies.

Program 11:

Write a Java Servlet program to demonstrate the use of GET and POST methods for handling HTTP client requests and server responses.

Program 12:

Write a JSP program to demonstrate the use of Java Beans.