Lab 11

Method and Recursion Tracing



CSE110: Programming Language I

| **No of Tasks** | | | **Points to Score**  **Classwork**  **2\*10 = 20** |
| --- | --- | --- | --- |
| **Classwork** | **Evaluation** | **Practice Problems** |
| **3** | **0** | **3** |

The students must complete the classwork tasks in the lab class to obtain the lab performance marks. They will also be marked based on the assessment tasks. The lab instructors may show/explain a few of the classwork tasks to the students if necessary. Any plagiarism in classwork or homework will lead to the student getting zero in the entire assignment. A random viva may take place.

You must not hard code any of the tasks, which means your code should work for any valid user input.

**Classwork 1**

**Trace the following code to generate the outputs. Show the necessary trace table.**

| **1** | **class ClassWork1{** | **OUTPUT** |
| --- | --- | --- |
| **2** | **public static int f3(int a, int c){** |  |
| **3** | **System.out.println("F3 begins");** |  |
| **4** | **System.out.println(a+c);** |  |
| **5** | **System.out.println("F3 ends soon");** |  |
| **6** | **return a\*5;** |  |
| **7** | **}** |  |
| **8** | **public static String f1(int n){** |  |
| **9** | **System.out.println("F1 begins");** |  |
| **10** | **f2(5,n);** |  |
| **11** | **System.out.println("F1 ends soon");** |  |
| **12** | **return n+1+" from F1";** |  |
| **13** | **}** |  |
| **14** | **public static void main(String[] args){** |
| **15** | **System.out.println("Starting soon...");** |
| **16** | **int c = 99;** |
| **17** | **String s = f1(c);** |
| **18** | **System.out.println(s);** |
| **19** | **System.out.println("The End");** |
| **20** | **}** |
| **21** | **public static void f2(int c, int d){** |
| **22** | **System.out.println("F2 begins");** |
| **23** | **System.out.println( f3(c+1,d-1) );** |
| **24** | **System.out.println("F2 ends");** |
| **25** | **}** |
| **26** | **}** |

**Classwork 2**

**Trace the following code to generate the outputs. Show the necessary trace table.**

| **1** | **class ClassWork2{** | **OUTPUT** |
| --- | --- | --- |
| **2** | **public static String fun(String s, int n){** |  |
| **3** | **if(s.length()==4){** |  |
| **4** | **return n+s+n;** |  |
| **5** | **} else if(n%2==0){** |  |
| **6** | **System.out.println(s+n+n+3);** |  |
| **7** | **return fun(s+n, n+3);** |  |
| **8** | **} else {** |
| **9** | **System.out.println(s+n+(n-1));** |
| **10** | **return fun(s+n, n-1);** |
| **11** | **}** |
| **12** | **}** |
| **13** | **public static void main(String[] args){** |
| **14** | **String s = fun("",1);** |
| **15** | **System.out.println(s);** |
| **16** | **}** |
| **17** | **}** |

# **Classwork 3**

Write a Java code that reads two files called **names.txt** and **jobs.txt** which contains names and their corresponding job titles respectively. The program will further combine the name with the corresponding job description as shown below and write them in a new file called **namesJobs.txt** .

Note: The number of names and jobs will always be equal.

The **names.txt** and **jobs.txt** file can be found from [**this link**](https://drive.google.com/drive/folders/1MjMhB7Y356O7FmP4ucUPegeoGr2mDUdT?usp=drive_link). You can download the files from the link and use it for your code. Moreover, File I/O slides can be found [**from here**](https://docs.google.com/presentation/d/1uqH1gbguXVc9d9o12CZ2oZFMlhb-2pwOkM3Idw50uL0/edit?usp=sharing).

| Sample **names.txt** file | Sample **jobs.txt** file | Sample **namesJobs.txt** file |
| --- | --- | --- |
|  |  |  |

# 

# Practice Problems **(No Submission)**

**Task 1**

| **1** | **public class P1{** |
| --- | --- |
| **2** | **public static void main (String args[]){** |
| **3** | **int a = 4, b = 7;** |
| **4** | **System.out.println(methodA(a,b));** |
| **5** | **}** |
| **6** | **public static double methodA(int m, int n){** |
| **7** | **int p = m+n-23, s = 0;** |
| **8** | **if (p<0){** |
| **9** | **System.out.println(p);** |
| **10** | **System.out.println(methodB(p+10));** |
| **11** | **s = methodB(p-10);** |
| **12** | **}** |
| **13** | **System.out.println(--s);** |
| **14** | **return p\*m+s;** |
| **15** | **}** |
| **16** | **public static int methodB(int r){** |
| **17** | **int q = 6;** |
| **18** | **System.out.println(++r+q);** |
| **19** | **return q-r;** |
| **20** | **}** |
| **21** | **}** |

**Task 2**

| **1** | **public class P2{** |
| --- | --- |
| **2** | **public static boolean met1(int n1, int n2){** |
| **3** | **System.out.println("Method 1");** |
| **4** | **int n = n1+n2;** |
| **5** | **System.out.println(n);** |
| **6** | **return met3(n, n2)>n1;** |
| **7** | **}** |
| **8** | **public static int met2(int n, String s){** |
| **9** | **int p = 5;** |
| **10** | **System.out.println("Method 2");** |
| **11** | **met1(n,p);** |
| **12** | **return s.length();** |
| **13** | **}** |
| **14** | **public static double met3(int n, int p){** |
| **15** | **System.out.println("Method 3");** |
| **16** | **System.out.println(n/p);** |
| **17** | **return p;** |
| **18** | **}** |
| **19** | **public static void main (String args[]){** |
| **20** | **System.out.println("Main Method");** |
| **21** | **System.out.println(met2(6,"ABC"));** |
| **22** | **}** |
| **23** | **}** |

**Task 3**

| **1** | **public class P3{** |
| --- | --- |
| **2** | **public static int calculate(int n) {** |
| **3** | **if (n <= 0){** |
| **4** | **return 4;** |
| **5** | **}** |
| **6** | **else if (n % 2 != 0){** |
| **7** | **return n + calculate(n - 1);** |
| **8** | **}** |
| **9** | **else{** |
| **10** | **return n \* calculate(n - 2);** |
| **11** | **}** |
| **12** | **}** |
| **13** | **public static void main(String[] args) {** |
| **14** | **int result = calculate(8);** |
| **15** | **System.out.println(result);** |
| **16** | **}** |
| **17** | **}** |