COMP2026 Problem Solving Using Object Oriented Programming

# Laboratory 13

**Part A Discovery Exercises**

**Task 1: Lambda Expression**

1. Given the following interface.

|  |
| --- |
| public interface NumChecker {  public boolean check(int n);  } |

Write lambda expressions to check whether the given integer is

1. divisible by 3
2. in between -5 and 10 inclusive

|  |
| --- |
| public class MyMainClass {  public static void main(String[] args){  new MyMainClass().runApp();  }  /\*\*  \* Print all the elements in the given integer array that  \* pass the check  \*  \* @param a - an integer array  \* @param c - a NumberChecker  \*/  public void printElements(int[] a, NumChecker c){  for (int i = 0; i < a.length; i++){  if(c.check(a[i])){  System.out.print(a[i] + " ");  }  }  System.out.println();  }  public void runApp(){  int[] intAry = {34, 6, 21, -1, -32, 24, -97, 76, 9};  //Example  System.out.print("Positive Elements: ");  printElements(intAry,(n)->{return n > 0;});  //your code goes here...  System.*out*.print("Divisible by 3 Elements: "); printElements(intAry,(n)->{return n % 3 ==0;});  System.*out*.print("In between -5 and 10 inclusive Elements: "); printElements(intAry,(n)->{return n >= -5 && n <= 10;});  }  } |

1. Given the following interface.

|  |
| --- |
| public interface ArrayAnalyzer {  double getResult(double[] a);  } |

Write a lambda expression to return maximum element in the given array.

|  |
| --- |
| public class MyArrayProg {  public static void main(String[] args){  new MyArrayProg().runApp();  }  /\*\*  \* Print the result of the ArrayAnalyzer  \*  \* @param a - a double array with size > 0  \* @param analyzer - an ArrayAnalyzer  \*/  public void printResult(double[] a, ArrayAnalyzer analyzer){  System.out.println(analyzer.getResult(a));  }  public void runApp(){  double[] array = {3.5, 54, 76.8, 48, 9.7, 8, 7};  //Example  System.out.print("Total: ");  printResult(array, (a)->{  double total = 0;  for (int i = 0; i < a.length; i++) {  total += a[i];  }  return total;  });  //your code goes here...  System.*out*.print("Maximum: "); printResult(array, (a)->{  double total = 0;  for (int i = 0; i < a.length; i++) {  if(a[i]>=total){  total = a[i];  }  }  return total; });  }  } |

**References**

1. Bravaco, R., & Simonson, C. (2009). *Java programming: From the ground up*. *Dubuque*, IA: McGraw-Hill.
2. Dean, J., & Dean, R. (2008). *Introduction to programming with Java: A problem solving approach*. Boston: McGraw-Hill.
3. Farrell, J. (2012). *Java programming. Boston, MA: Course Technology Cengage Learning*
4. Forouzan, B. A., & Gilberg, R. F. (2007). *Computer science: A structured programming approach using C (3rd ed.)*. Boston, MA: Thomson Course Technology.
5. Gaddis, T. (2016). *Starting out with Java (6th ed.)*. Pearson.
6. Liang, Y. D. (2013). *Introduction to Java programming: Comprehensive version*. (8th ed.). Pearson.
7. Schildt, H. ( 2006). *Java a beginner's guide*. New York: McGraw Hill.
8. Schildt, H., & Skrien, D. J. (2013). Java programming: A comprehensive introduction. New York: McGraw-Hill.
9. Wu, C. T. (2010). *An introduction to object-oriented programming with Java*. Boston: McGraw Hill Higher Education
10. Xavier, C. (2011). *Java programming: A practical approach*. New Delhi: Tata McGraw Hill.
11. yet another insignificant Programming Notes. (n.d.). Retrieved from https://www3.ntu.edu.sg/home/ehchua/programming
12. Zakhour, S., Kannan, S., & Gallardo, R. (2013). *The Java tutorial: A short course on the basics (5th ed.)*.
13. Edpresso Team. (2019, July 1). What is a Java lambda function? Educative: Interactive Courses for Software Developers. https://www.educative.io/edpresso/what-is-a-java-lambda-function