CS 330 – Programming Languages
HW1F

NAME _______Stuart Lech________ SID __800728996____

Give concise answer to the following questions based on materials discussed in PLP Chapter 7, both textbook and notes. **Save your doc as PDF and submit it on Moodle. This HW is worth 50 points (12 for each question and 2 for save as PDF) and is due Wednesday (4/10) at 10 PM. Assignment will be graded both on effort and correctness.**

1. Briefly discuss 2 types of parameters used in subroutines and give an example for each in a language of your choice.
   Call by Value: This method copies the actual value of an argument into the formal parameter.
   Changes inside the subroutine do not affect the actual parameter. For example, in Python:

```
 def increment(a):
 a += 1
 print("Inside function: ", a)
 x = 5
 increment(x)
 print("Outside function: ", x)
```

   Call by Reference: Copies the address of an argument into the formal parameter. Changes affect
   the actual argument. For example, in C++:

```
 void increment(int &a) {
 a += 1;
 cout << "Inside function: " << a << endl;
 }
 int main() {
 int x = 5;
 increment(x);
 cout << "Outside function: " << x << endl;
 return 0;
 }
```

2. Briefly describe exception handling and give an example in a language of your choice.
   Exception handling is a method to deal with unexpected errors, allowing separation of error handling code. For example, in Java:

```
public class Example {
 public static void main(String[] args) {
 try {
 int divideByZero = 5 / 0;
 } catch (ArithmeticException e) {
 System.out.println("An arithmetic exception occurred: " + e.getMessage());
 } finally {
 System.out.println("This block is always executed.");
 }
}
}
```

3. Briefly describe two methods for heap management pertaining to allocation of single-size memory cells.

Bitmaps: Uses a bitmap to track allocated and free cells, allowing quick allocation but may suffer from fragmentation.

Free Lists: Maintains a list of free memory cells, efficient in memory usage but may be slower in allocation and deallocation.

4. Briefly discuss similarity and difference between struct and union in C++.

Similarity: Both allow grouping of different data types.

Difference: `struct` allocates separate memory for each member allowing simultaneous access, whereas `union` shares the same memory among all its members, allowing only one member to contain a value at any given time.