

PPL LAB EXAM QUESTION SET

1. a. Examine lifetime and scope of static variables in C for the following:

- i) Accessing a static variable which has function scope
- ii) Accessing a static variable which has file scope
- iii) To examine the default value of the static variable. (-- use gdb to print default values for all possible data types, declared as static). Your code should check the initial values in case of function scope as well as in file scope for the static variable.
- iv) Can the static variable be externed? Show the global variable, declared with an extern storage class.
- v) Declaring the static variable in the header file and examining the behavior in multiple implementation files
- vi) Examining the scope of a static variable within a compilation unit
- vii) External and internal linkages
- viii) Check the usage of static functions in C.
- ix) Can a pointer to a static variable returned from a function? What happens to the scope of the variable then? Demonstrate.
- x) Demonstrate the differences between static variables and global variables.

b. Illustrate with suitable programs the concept of call resolution in Java. (prepare yourself to explain how it is different in C++, implementation not required.)

2. a. Examine garbage / memory leak in 'C'

- i. using memcheck.h library file.
- ii. using valgrind tool.

Develop a mechanism to avoid mem-leak.

b. Demonstrate shallow and deep copy of lists in python

demonstrate all possible ways of list copy. Show the type of copy (as deep or shallow) by the object ids and also modify one list and observe the list copy for the change.

- i. using assignment operator
- ii. Using the list() function and using the sub list(sublist copy or slicing) mechanism.
- iii. Nested lists
- iv. Use copy and deepcopy methods. Demonstrate the difference using nested list.

3. a. Examine assignment operations for

- i. Structures in 'C'
 - a) Variables of the same structure are they equivalent?
 - b) What happens when an employee type variable is assigned to emp type variable? Why?
 - c) When typedef is used in C, will it create a new type or only a type name?
 - d) If int is declared with typedef as myint, and there are two variables,(i1 – int and my1 – myint) , then comment about their compatibility.
 - e) Structure variable copy using assignment operator. Also, when structure consists of an array as a member.
 - f) Structure array variables copied using assignment operator?

ii. **Study Assignment of Arrays in Java:** demonstrate different ways of copy. In each case make an observation on the type of copy.

- a) Assignment operator.
- b) Using an ArrayList and then make a copy.
- c) For arrays, is there a deepCopy method available? If yes, demonstrate.
- d) What happens when objects are copied using assignment operator?
- e) How to make deep copies of objects?
- f) How to use arraycopy.
- g) Show the usage of == and .equals() to check the equality? How are they different.

b. Illustrate callbacks in Python

4. a. **Examine closures in Python. Demonstrate the closure-effect in 'C' using structures and function pointers.**
b. **Demonstrate usage of variable number of arguments in Java and Python.**
c. **Demonstrate deep copy and shallow copy for object assignments in Java**
5. **Demonstrate the usage of wild card parameters in Java generics. Can we impose any restrictions regarding the type of objects passed for the wildcard parameters? If yes, demonstrate the same.**
b. **Demonstrate usage of variable number of arguments with user defined functions in C.**

a.
6. a. **Demonstrate with a Java program, the way you would implement a generic method to compare:**
 - i. Any basic types (autoboxing)
 - ii. Any two objects. (of user defined class)b. **Illustrate tail recursion in 'C'. (must follow all the methods that are done in regular lab)**
7. a. **Using Generic classes in Java, define a stack (that can contain any object). Provide push, pop and display methods for the stack defined. Demonstrate the working with a main method.**
b. **Write a Python program to illustrate Keyword Parameters.**
8. a. **Demonstrate multithreading in Java using 3 threads. Show the racing among threads by using a class variable that gets modified in the run methods.**
 - Develop a mechanism to overcome racing and achieve synchronization among the threads (use a synchronized method or a synchronized block)b. **Accomplish an unconditional jump across functions within a single file in C program.**
9. a. **Create 3 threads, (in Java)**
 - Show the execution sequence of the four threads including main.(use proper print lines)
 - Demonstrate the usage of join and sleep methods to alter this execution sequence.b. **using interfaces illustrate the implementation of multiple inheritance in Java.**
10. a. **Demonstrate the working of the following Java collections:**
 - i. Lists
 - ii. Set

iii. Map

b. Demonstrate stack smashing in C, how can it be avoided? How can we suppress the compiler from detecting this?

i. a. Examine assignment operations for

Structures in 'C'

- g) Variables of the same structure are they equivalent?
- h) What happens when an employee type variable is assigned to emp type variable? Why?
- i) When typedef is used in C, will it create a new type or only a type name?
- j) If int is declared with typedef as myint, and there are two variables, (i1 – int and my1 – myint) , then comment about their compatibility.
- k) Structure variable copy using assignment operator. Also, when structure consists of an array as a member.
- l) Structure array variables copied using assignment operator?

ii. Study Assignment of Arrays in Java: demonstrate different ways of copy. In each case make an observation on the type of copy.

- h) Assignment operator.
- i) Using an ArrayList and then make a copy.
- j) For arrays, is there a deepCopy method available? If yes, demonstrate.
- k) What happens when objects are copied using assignment operator?
- l) How to make deep copies of objects?
- m) How to use arraycopy.
- n) Show the usage of == and .equals() to check the equality? How are they different.

b. Demonstrate multiple inheritance in Python.

11. a. Examine the scope of goto statement in a C program for the following:

- i. Within a function
- ii. within a compilation unit
- iii. Across functions in the same file
- iv. Across files
- v. Forward and backward jumps.

b. Demonstrate usage of variable number of arguments in Java and Python.