CSE 425: Concepts of Programming Languages

Summer 2020, Faculty – ARa2

Midterm, Marks: 65, Time: 3 hours

Date: ____________

(Use papers to write your answers by hand. Answer serially. Make PDF of **clear** photos of those papers and upload that in google classroom by the deadline. Sharers/copiers of any answer will be given zero.)

Descriptive Questions (Answer Q1 and any 1 question from Q2-4) [7*2=14 points]

1. Differentiate among different programming language paradigms and discuss their relative advantages and disadvantages. Explain their differences using a code snippet of a representative language of each paradigm. Also discuss their typology with examples.

2. What does it mean by type binding? Discuss the ways type binding happens in different programming languages along with examples. Also discuss their advantages and disadvantages.

3. What is the scope of a variable? Discuss classification of variables (along with examples) according to their scopes as well as mention the names of programming languages supporting that type of scope. Also discuss their advantages and disadvantages.

4. What is InOut model of parameter passing? Discuss different ways to implement this type of parameter passing along with their usages in different languages. Also discuss their advantages and disadvantages.

Compute Output (Answer all, you must explain why you got this output in details to get marks): [6+8 = 14 points]

5. Find the output of the following Python program and explain why are you getting that output by elaborating what changes are happening to each variable in each line.

```
def sub1(a):
    a = [0,1,2,3,4,5,6]
    a[0:3] = [9,8,7]

def sub2(a):
    a[0:3] = [9,8,7]
    a = [0,1,2,3,4,5,6]

x = [10,11,12,13,14,15,16]
sub1(x)
print(x)
sub2(x)
print(x)
```

6. Find the output of either of the following CPP programs (either (a) or (b)) and explain why are you getting that output by elaborating what changes are happening to each variable in each line.

(a)

```
#include<iostream>
using namespace std;

int f(int &x){
   x = x/10;
   return x;
}

int g(){
   static int k = 123;
   int r = 1+f(k);
   return r;
}

int main(){
   cout<<g()<<endl;
   cout<<g()<<endl;
   cout<<g()<<endl;
   return 0;
}
```

(b)

```
#include <iostream>
using namespace std;
int count = 3;
void sub(int count=10) {
 while (count) {
   if(count > 6){
      cout << count-- <<endl;
      continue;
   }
   else if(count >= 2){
      int count = --::count;
      cout << count<<endl;
      if(!count) break;
   }
   count -= 2;
 }
}

int main(){
   sub(10);
   cout<<count;
   return 0;
}
```

Correcting Program [You must explain the issues in details to get marks] [14 points]

7. [5+4+5=14 pts.] Find bugs/errors present in both of the following two programs and explain why they may cause any issue. Rewrite (again, using a union) the program (i) by fixing all the issues in it.

(i)

```
//This is a C program
union Test {
   int id;
   char name[10];
};

void main(){
   Test t;
   printf("Enter your ID and name: ");
   scanf("%d%s", t.id, t.name);
   printf("You entered: %d, %s", t.id, t.name);
}
```

(ii)

```
//This is a CPP program
int *q;
class A{
   int *p;
public:
   A(){ p = new int;}
   ~A(){delete p;}
   void save(){q = p;}
};
int main(){
   int x;
   cin>>x;
   if(x == 1){
      A a;
      a.save();
   }
   *q = 5;
   return 0;
}
```

Write Program [Answer either (a) or (b) in Q8 – 11] [3+4+6+10 = 23 points]

8. a) Write a code-snippet in *Scheme* that evaluates the mathematical expression: 41*9+5*23/7 **OR**

   b) Write a single-line code in *Python* that creates a list containing the square of all the proper divisors of a number x (assume that the value of x is already read from user). For e.g. if x = 35 then the list should be [25, 49].

9. a) Define a function in *Scheme* that takes two parameters: x and y, and which returns – (i) "Greater" if x is greater than y, (ii) "Less" if x is less than y, and (iii) "Equal" otherwise. **OR**

   b) Write a *Bash* program that takes two arguments and prints "Greater", "Less", or "Equal" according to the logic in the alternative question # (a).

10. a) Define a recursive function in *Scheme* that takes a parameter: *n* and which evaluates to the sum of the series: $\frac{2}{3}+\frac{3}{4}+\frac{4}{5}+\cdots up\ to\ n'th\ term.$ **OR**

    b) Write a program in *Bash* that takes an integer argument, say n, and then prints names of all the files whose size is greater than n. You can only use the commands I taught in the class for this purpose.

11. Write a (a) *Perl or (b) Bash* program that takes two arguments: a positive integer and a file-path. If the 1st argument is not positive or 2nd argument is not a valid file-path, the program should exit with an error. Otherwise, it should find all the "special-prime" numbers (whose sum of digits is prime) below the 1st argument (e.g. all the special-primes below 100, when the 1st argument is 100) and then write them into a file indicated by the 2nd argument. For e.g. if your program's first argument is 17 and second argument is "/home/abc/file.txt", then your program should write the following numbers in "/home/abc/file.txt": 2,3,5,7,11,12,16