ROLL NO:	

PRESIDENCY UNIVERSITY, BENGALURU

SCHOOL OF ENGINEERING

GAIN MORE KNOWLEDGE
REACH GREATER HEIGHTS

Weightage: 20 % Max Marks: 40 Max Time: 1 hr. Saturday, 29th September, 2018

TEST – 1 **SET B**

Odd Semester 2018-19 Course: **CSE-214 Principles of Programming Language** V Sem. Computer Science

Instruction:

(i) Read the question properly and answer accordingly.
(ii) Question paper consists of 3 parts.

Part A

(2 Q x 6 M = 12 Marks)

Q1. The new datatype called TIME is introduced in the new programming language which is being designed. The TIME is combination of Hour, Minute, and Second.

i. Design the syntax to read the value for time. (Similar to scanf statement in C Language)
ii. Design the syntax to display the value of time. (Similar to printf statement in C Language)

i.e If the declaration of time is **TIME t1;** Give the syntax to read and display the value for t1. Discuss and justify the design in user and compiler perspective.

Note: Do not design the grammar..

Q2. Discuss memory requirements of the program written in C like programming language in Figure1.

int fun3()
{
int x,y,z;
------ 	
return y;
}
	void fun2(int a, int b)
	{
	static int c,d;
	int *ptr,x;
	---------- 	
	Ptr=(int)malloc(c*sizeof(int));
	X=fun3();
	}
		void fun1()
		{
		int a,b;
		fun2(a,b);
		.
		------ 	
		}
			float x,y
			void main()
			{
			float s,t;
			fun1();
			}

Figure 1

Part B

(1 Q x 12 M = 12 Marks)

Q3. Explain the run time stack for the program in Figure 1, when function fun3 is executing.

Note: Start the frame number of main with the address 2000

i) Deep access implementation of runtime stack.
ii) Shallow access implementation of runtime stack.

Part C

(1Q x 16 M = 16 Marks)

4. Consider the grammar of a Programming Language: The non terminals are enclosed within <NT> brackets, others are terminals, and <Program > is the starting symbol.
The Productions rules are given below.

1) *<Program>* →*<functions>*comma*<functionbody>*comma
2) *<functions>*→*<function><functions>*/ ε
3) *<function>* →*<funsignature><functionbody>*
4) *<funsignature>*→ *<type>*< id> (*<params>*)
5) *<type>*→ int / float/string
6) *<params>*→ *<type>*< id> comma *<params>* / ε
7) *<functionbody>*→{ *<declarations>* *<statements>* *<E>; }*
8) *<declarations>*→ *<type>*<id> ; <declarations >/ ε
9) *<statements>* →< id> =*<E>*;<statements>/ <id>=<id><more>;<statements>/ε
10) *<E>*→ *<E>+<E >*/ *<E>*<E>*/*<id>*/*<intigerliteral>*/*<floatliteral>*/*<stringliteral>*
11) *<more>* →(*<args>*)
12) *<args>* →< id> , *<args>* / ε
13) <id>→ [a-z]$^+$ [A-Z] [0-9]*[a-z]*[A-Z]*
14) *<integerliteral>*→[0-9]$^+$
15) *<floatliteral>*→[0-9]*.[0-9]$^+$
16) *<stringliteral>*→[a-z]$^+$

a) Consider the following program. Prove that the program belongs to the grammar.
Hint: Show the derivation.

```
comma
{
int a;
float b;
a =a*b+a+1;
*a;
}
comma
```

b) Consider the following program. Prove that the program does not belong to the grammar.
Hint: Show the derivation.

```
comma
{
int a;
float b;
a = fun1(a,);
}
comma
```

ROLL NO:	

PRESIDENCY UNIVERSITY, BENGALURU

SCHOOL OF ENGINEERING

Weightage: 20 %   Max Marks: 40   Max Time: 1 hr.   Saturday, 29th September, 2018

TEST – 1   **SET A**

Odd Semester 2018-19   Course: **CSE-214 Principles of Programming Language**   V Sem. Computer Science

Instruction:

(i) Read the question properly and answer accordingly.
(ii) Question paper consists of 3 parts.

Part A

(2 Q x 6 M = 12 Marks)

Q1. The new datatype called TIME is introduced in the new programming language which is being designed. The TIME is combination of Hour, Minute, and Second. Design the syntax of comparison operator to compare two variables of type TIME.

i.e If the declaration of time is **TIME t1, t2;** Give the syntax to compare the value of t1 and t2.

Discuss and justify your design in terms of readability and writability in user perspective and writability in terms of compiler perspective.

Note: Do not design the grammar.

Q2. Discuss memory requirements of the program written in C like programming language in Figure1.

void function3()
{
int x;
------ 	
}
	int function1(int a, string b, int s)
	{
	static int c,d;
	function3();
	....
	return(a);
	}
		function2(float a, int b)
		{
		int *ptr, *str, x,y,z;
		
		x=function1(a,b);
		....
		}
			float m,n;
			
			void main()
			{
			int ar[10], *at,n,a,b;
			scanf("%d",&n);
			at=(int)malloc(n*sizeof(int));
			function2(n,n);
			}

Figure 1

Part B

(1 Q x 12 M = 12 Marks)

Q3. Explain the run time stack for the program in Figure 1, when function3 is executing.
Note: Start the frame number of main with the address 1000
i) Deep access implementation of runtime stack.
ii) Shallow access implementation of runtime stack.

Part C

(1Q x 16 M = 16 Marks)

4. Consider the grammar of a Programming Language: The non terminals are enclosed within <NT> brackets, others are terminals, and <Program > is the starting symbol.

The Productions rules are given below.

1) *<Program>* →*<functions>*comma*<functionbody>*comma
2) *<functions>*→*<function><functions>/ ε*
3) *<function>* →*<funsignature><functionbody>*
4) *<funsignature>*→ *<type>*< id> (*<params>*)
5) *<type>*→ int / float/string
6) *<params>*→ *<type>*< id> comma*<params>* / ε
7) *<functionbody>*→{ *<declarations> <statements>* **<E>*; }
8) *<declarations>*→ *<type>*<id> ; <declarations >/ ε
9) *<statements>* →< id> =*<E>*;<statements>/ <id>=<id><more>;<statements>/ε
10) *<E>*→ *<E>+<E >/ <E>***<E>/<id>/<intigerliteral>/<floatliteral>/<stringliteral>*
11) *<more>* →*(<args>)*
12) *<args>* →< id> , *<args>* / ε
13) <id>→ [a-z]$^{+}$ [A-Z] [0-9]*[a-z]*[A-Z]*
14) *<integerliteral>*→[0-9]$^{+}$
15) *<floatliteral>*→[0-9]*.[0-9]$^{+}$
16) *<stringliteral>* →[a-z]$^{+}$

a) Consider the following program. Prove that the program belongs to the grammar.
Hint: Show the derivation.

```
comma
{
int a;
a = a+big*small;
*a;
}
comma
```

b) Consider the following program. Prove that the program does not belong to the grammar.
Hint: Show the derivation.

```
float sss(int a,)
{
a= a+1;
}
comma
{
int a;
a = fun1(a,);
*z;
}
comma
```