

Name: .....

All statements should be written in C++.

You may find a partial ASCII table at the end of this document.

1. [4 pts] Write the preprocessor directive that you would need to use for the followings.

I. `cout << "What is your initials?";` # include < .....>

II. `string name` # include < .....>

III. `abs(x)` # include < .....>

IV. `setprecision(2)` # include < .....>

2. [4 pts] Distinguish each of the variable name if it is legal or illegal. If illegal, explain why.

	Variable name		
		Legal?(Y/N)	If illegal, why?
I.	tax amount		
II.	taxAmount		
III.	2018tax		
IV.	tax2018		
V.	tax\$		

3. [2.5 pts] Identify the validity of the following statement(s). If a statement is invalid, explain why not.

	Statement(s)	Validity	
		Valid/ invalid	If invalid, why?
I.	<code>const double MOR_RATE = 4.27;</code>		
II.	<code>const double MOR_RATE = 4.27;</code> <code>MOR_RATE = 3.75;</code>		
III.	<code>const double MOR_RATE;</code>		

4. [4 pts] Which of the following are valid character literals?

	literal	Character Literal (Y/N)	if not a character literal, why?
I.	68		
II.	"C"		
III.	'c'		
IV.	"\0"		
V.	'B+'		

5. [2 pts] You are to write a program to maintain the student scores of a class. Write statements to do each of the followings:

- I. Define a variable with a proper type and a proper name for the number of students in this class.
- II. Assign 30 to the variable.

.....

.....

.....

6. [4 pts] If **income** and **expenditure** are numeric variables, and **companyName** is a string variable, which of the following statements are valid assignments? If a statement is not valid, explain why not.

	Statement	Validity (Y/N)	If invalid, why?
I.	<i>income</i> = 100;		
II.	<i>expenditure</i> = "2400.99";		
III.	<i>income</i> = 31,305.52;		
IV.	<i>companyName</i> = "ABC company";		
V.	<i>income</i> * 2.1 = <i>expenditure</i> ;		

7. [3 pts] Write ONE statement that consolidate the following statements.

```
int flightNum = 89;
int travelTime;
int departure = 10;
int distance;
```

.....

.....

.....

.....

.....

8. [2 pts] Write down a complete C++ initialization statement to store a letter grade. Name the variable **grade** and set the value to be **A**.

.....

.....

9. [3 pts] Complete the following table by calculating and writing the value of each C++ expression

Expression	Value
<code>x = 5 + 2 * 4 + 10 / 5 - 3</code>	
<code>x = (5 + 2) * 4 + 10 / (5 - 3)</code>	
<code>x = 4 + 2 * (6 - 2) + (4 + 17) % 2 - 1</code>	

10. [2 pts] Convert the following Algebraic expressions to programming statement (you may assume the variable definition has been done priori):

$$\text{distance} = p(5rs + \frac{4}{3}r^{30})$$

.....

.....

.....

.....

11. [2.5 pts] Assume the following variable definitions and predict the output.

```
short LuckyNum = 2;
char initials = 'R';
```

Statement	Output
<code>cout &lt;&lt; sizeof(initials) &lt;&lt; endl;</code>	
<code>cout &lt;&lt; sizeof(LuckyNum) &lt;&lt; endl;</code>	
<code>cout &lt;&lt; sizeof(2 * LuckyNum) &lt;&lt; endl;</code>	
<code>cout &lt;&lt; sizeof(2 * 1.0) &lt;&lt; endl;</code>	
<code>cout &lt;&lt; sizeof(2 * initials) &lt;&lt; endl;</code>	

12. [5 pts] Assume `int x = 6, a = 2, b = 3, c = 1, d = 2`. Find the value of `x` after the statement.

Statement	Value of x
<code>x += 4 + b;</code>	
<code>x -= 3 * d;</code>	
<code>x *= 11 - c;</code>	
<code>x /= b;</code>	
<code>x %= d;</code>	

What would be the output of the each of the following sample codes; Assume individual codes were embedded in an error free program (#13 - #22 - 2 pts each).

13.

```
cout << "Blue " << endl << "Green" << endl << "Red";
```

output:

```
.....
.....
.....
.....
```

14.

```
cout << "one " << "two " << "Three";
```

output:

```
.....  
.....  
.....  
.....
```

15.

```
cout << "Enter the last three temperature readings." << endl;  
cin >> temp3 >> temp2 >> temp1;  
cout << temp1 << endl << temp2 << endl << temp3 << endl;
```

Output:

```
Enter the last three temperature readings.  
101 100 98 [Enter]
```

```
.....  
.....  
.....  
.....
```

16.

```
string school = "Webster";  
cout << "My School is " << "school";
```

output:

```
.....  
.....
```

17.

```
double number = 7.5;  
cout << number << endl;
```

output:

.....  
.....

18.

```
double tmpNumber = 7.5;  
int number = tmpNumber;  
cout << number << endl;
```

output:

.....  
.....

19.

```
double ratio;  
ratio = 5 / 2;  
  
cout << ratio << endl;
```

output:

.....  
.....

20.

```
double ratio;  
ratio = 5.0 / 2.0;  
  
cout << ratio << endl;
```

output:

.....

21.

```
int ratio;  
ratio = 5 % 2;  
  
cout << ratio << endl;
```

output:

.....  
.....

22. [3 pts] Write the output.

Statement	output
cout << 11/ static_cast <float> (2);	
cout << static_cast <char> (66);	
cout << static_cast <float> (11/2);	

23. [2 pts] Write the output of the following statement in the given space.

```
double distance = 62.3472;
```

```
cout << fixed << setprecision(2) << distance << endl  
      << scientific << setprecision(3) << distance << endl;
```

output:

.....  
.....  
.....

24. [3 pts] Write the output of each statement.

Statement	Output
cout << pow(2.0, 3.0);	
cout << abs(-6);	
cout << sqrt(4.0);	

25. [2 pts] Fill in the blank with a single statement that reads the user input given in one-line (as in the output: *470 Lockwood Ave*) and store it in the variable *address*.

```
string address;
cout << "Enter the address";

.....

.....

.....
```

Output:

*Enter the address*

**470 Lockwood Ave**[Enter]

26. [2 pts] Write a statement to read a character, including white characters, from the keyboard and store it in the variable *continue*.

```
char continue;
cout << "Press any key to continue....";

.....

.....
```

27. [4 pts] Write a C++ statement to get the following output.

You may NOT use any spaces or tabs to align the output. Use only stream manipulators: *setw()*, *fixed*, *setprecision()*, *left*, and *right*. Note: The borders and the serial numbers are just to make it easier to read; they would not be displayed in the output. Also assume the following statements already have been made.

```
double price = 2.75
count = 13,
total;
total = price * count;
```

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
G	a	l	l	o	n		P	r	i	c	e			2	.	7	5
G	a	l	l	o	n		C	o	u	n	t					1	3
T	o	t	a	l									3	5	.	7	5

```
.....

.....

.....

.....

.....
```







[illegible]

Printable ASCII Characters			
Dec	Hex	Oct	Character
65	41	101	A
66	42	102	B
67	43	103	C
68	44	104	D
69	45	105	E
70	46	106	F
71	47	107	G
72	48	110	H
73	49	111	I
74	4a	112	J
75	4b	113	K
76	4c	114	L
77	4d	115	M
78	4e	116	N
79	4f	117	O
80	50	120	P
81	51	121	Q
82	52	122	R
83	53	123	S
84	54	124	T
85	55	125	U
86	56	126	V
87	57	127	W
88	58	130	X
89	59	131	Y
90	5a	132	Z
91	5b	133	[
92	5c	134	\
93	5d	135	]
94	5e	136	^
95	5f	137	_

Printable ASCII Characters			
Dec	Hex	Oct	Character
96	60	140	`
97	61	141	a
98	62	142	b
99	63	143	c
100	64	144	d
101	65	145	e
102	66	146	f
103	67	147	g
104	68	150	h
105	69	151	i
106	6a	152	j
107	6b	153	k
108	6c	154	l
109	6d	155	m
110	6e	156	n
111	6f	157	o
112	70	160	p
113	71	161	q
114	72	162	r
115	73	163	s
116	74	164	t
117	75	165	u
118	76	166	v
119	77	167	w
120	78	170	x
121	79	171	y
122	7a	172	z
123	7b	173	{
124	7c	174	
125	7d	175	}
126	7e	176	~