CSC133
REVIEW FOR FINAL

1. Compute the value of each of the following expressions or write INVALID

a. 20 % 4
b. 20/4
c. 8 % 16 + 16/8
d. 5/(4%4)
e. 3.0 % 2

2. Write the following expressions in valid C++ statements

a. $\frac{a + d * c}{2} + b$

b. $\frac{a + b}{c}$

3. If int1 and int2 are int variables and real1 is a float show the resulting output using b to indicate each blank

```
int1=45;
int2=89;
real1=13.667;
cout<< fixed << showpoint;
cout<<setw(5)<<int1<<endl;
cout<<setw(2)<<int2<<endl;
cout<<setw(1)<<int1<<endl;
cout<<int1<<endl;
cout<<setprecision(1)<<setw(5)<<real1<<endl;
```

4. Evaluate each of the following logical expressions and indicate if they are TRUE (T) or FALSE (F).

a. (5>2) && !(2==5)   T   F

b. (5>2) || !(2==5)   T   F

c. 3!=3 && 7<8 || 2<3   T   F

5. What will the following code segments print?

a.

```
int1=120;
int2=30;
if ((int1>100) && (int2 =50))        // Assignment used instead of equality
  int3 = int1 + int2;
else
  int3= int1 - int2;
cout<<int1<<" "<<int2<<" "<<int3;
```

b.

```
angle=0;
if (angle > 5)
  angle = angle * 5;
else
  if (angle > 2)
     angle = angle + 10;
cout<<angle;
```

c.

```
count=1;
while (count <=4)
 {
       cout << count<<endl;
       count++;
 }
```

d.

```
count=1;
while (count <=4)
       cout << count<<endl;
       count++;
```

e.

```
count=1;
while (count <=4)
  {
       count++;
       cout << count<<endl;
  }
```

6. What is the output of the following code?

```
void Test (int alpha)
{
   static int n=5;

   n = n + alpha;
   cout<<n<<endl;
}

void main()
{
  Test(20);
  Test(30);
}
```

7. Given the function definition

```
int Mystery(float someVal)
{
   if (someVal > 2.0)
       return 3 * int(someVal);
   else
       return 0;
}
```

What is the value of the expression Mystery(4.2)?

8. What is the output of the following code?

```
int e = 5;
int main()
{
  int c,d;
  void change(int, int&);

  c=4;
  d=17;
  change(c,d);
  cout<<c<<"  "<<d<<"  "<<e<<endl;
}

void change(int p, int& q)
{
  int c;

  c = q;
  q = p + c;
  p = e;
  cout<<c<<"  "<<e<<"  "<<p<<"  "<<q<<endl;
}
```

9. Why does the following Switch statement cause a compile-time error?

```
switch (n)
{
        case 6: alpha = 10;
                break;
        case 2:
        case 5: alpha = 20;
                break;
        case 8:
        case 2:  alpha = 30;
}
```

10. What is the output of the following code fragment.

```
int1=4;
switch (int1)
{
        case 1:  cout<<1<<endl;
                 break;
        case 2:  cout<<2<<endl;
                 break;
        case 3:  cout<<3<<endl;
                 break;
        case 4: cout<<4<<endl;
        case 5: cout<<5<<endl;
}
```

11. What is the output of the following code fragment?

```
limit = 8;
cout<<"H";
loopcount = 10;
do
{
  cout<<"E";
  loopcount++:
}
while (loopcount <= limit);
cout << "LP";
```

12. Write the for loop which is equivalent to the following while loop.

```
count = -5;
sum=0;
while (count <=15)
{
  sum = sum + count;
  count++;
}
```

13. Write the for loop which will sum the integers between 0 and 100, inclusive.

14. Write the for loop which will read 25 integer values from the file "datain.dat". Compute the average of the values read.

15. Write a complete C++ program that reads a salesman's ID number and amount sold. Write out the ID, followed by "GETS AWARD" if amount sold is at least $50,000, "DOING OK" if amount sold is between $25,000 and $50,000 and "NEEDS PEP TALK" if amount sold is $25,000 or less.

16. Write a function that reads a file "datain.dat", of integers until a value of -100 is found. Return the average of the integers read.

17. Write a program segment that counts and displays the number of times a zero occurs in a file "datain.dat" of 50 integers.

18. Write a function that reads a file "datain.dat" of integer values representing temperatures and returns the highest temperature read.

19. Write a function SUMARRAYS that receives two one-dimensional parallel arrays LIST1 and LIST2 of integers, and the number of values in each array LENGTH, as parameters and returns the sum of the values in LIST1 plus the sum of the values in LIST2.

20. Write a function NEGATIVE that receives an array of integers, LIST, and the LENGTH of LIST as parameters and returns the index of the first negative number in array LIST. If there are no negative numbers in the array, return -1.

21. Write a segment of code to find and display the largest and smallest value in the integer array LIST of length LENGTH.

22. Write a segment of code to find and display the largest and smallest values in a two-dimensional M X N integer array TABLE.

23. Write the function FIND to return column of the first occurrence of 100 in the row ROW in a two-dimensional M X N integer array TABLE. If 100 does not occur in the row ROW, return -1.

24. Write the function FIND to return the row of the first occurrence of 100 in the column COL in a two-dimensional M X N integer array TABLE. If 100 does not occur in the column COL, return -1.

25. Write the function YES_OR_NO to return TRUE (1) if 100 occurs in the M X N integer array TABLE, and FALSE (0), otherwise.

REVIEW FOR FINAL - ANSWERS

1.
 a. 0
 b. 5
 c. 10
 d. invalid - cannot divide by zero
 e. invalid - MODULUS (%) can only have integer operands

2.
 a. (a+d*c)/2 + b
 b. (a+b)/c

3.
```
bbb45
89
45
45
b13.7
```

4.
 a. T
 b. T
 c. T

5.
a. 120 50 170 (= sign)
b. 0
c. 1
 2
 3
 4
d. infinite loop (no curly braces)
e. 2
 3
 4
 ~~4~~ 5

6. 25
 55

7. 12

8. 17 5 5 21
 4 21 5

9. The case 2 appears twice (default case not required)

10.
```
4
5
```

11.
```
HELP
```

12.
```
for(count=-5, sum=0; count<=15; count++)
        sum += count ;
```

13.
```
for(i=0, sum =0; i<=100; i ++)
    sum += i;
```

14.
```
ifstream inp;
inp.open(“datain.dat”);
for(i=1; i<=25; i++)
 {
        inp>>value;
        sum += value;
 }
average = sum/25;
```

15.
```
#include <iostream.h>
int main()
{
  int sales_ID;
  float amount_sold;

  cin>>sales_ID>>amount_sold;
  cout<<sales_ID<<“  “;
  if (amount_sold >= 50000)
        cout<<“GETS AWARD”<<endl;
  else
        if (amount_sold >25000)
                cout<<“DOING OK”<<endl;
        else
                cout<<“NEEDS PEP TALK”<<endl;
}
```

16.

```
 float AVERAGE(void)
 {
   ifstream inp;
   inp.open("datain.dat");

   int count=0, sum=0, value;

   inp>>value;
   while (value <> -100)
    {
        count++;
        sum += value;
        inp>>value;
    }
  if (count!=0)                              // or      if (count)
        return (float (sum)/ count);
  else
        return 0.0;
}
```

17.

```
int count=0, i, value;
ifstream inp;
inp.open("datain.dat");

for(i=1;i<=50;i++)
 {
        inp>>value:
        if (value ==0)                        // or    if (!value)
           count++;
 }
cout<<count<<endl;
```

18.

```
int function fhighest(void)
{
  ifstream inp;
  inp.open("datain.dat");
  int value, highest;

  highest = INT_MIN;                    // or   highest = some very low number  (-500)
  inp>>value;
  while (inp)
   {
       if (value>highest)
         highest = value;
       inp>>value;
   }
  return highest;
}
```

19.

```
int SUMARRAYS(int LIST1[], int LIST2 [], int LENGTH)
{
  int i, sum=0;
  for (i=0;i<LENGTH; i++)
        sum = sum + LIST1[i] + LIST2[i];
  return sum;
}
```

20.

```
int NEGATIVE(int LIST[], int LENGTH)
{
  int i, index = -1;
  Boolean found_neg=FALSE;

  for (i=0; i<LENGTH && !found_neg; i++)
        if (LIST[i] < 0)
         {
                index = i;
                found_neg=TRUE;
         }
  return index;
}
```

21,

```
largest = LIST[0];
smallest = LIST[0];
for (i=1;i<LENGTH;i++)
 {
        if (LIST[i] < smallest)
          smallest = LIST[i];
        if (LIST[i] > largest)
          largest = LIST[i];
 }
cout<<"The value of smallest in LIST is "<<smallest<<endl;
cout<<"The value of largest in LIST is "<<largest<<endl;
```

22.

```
largest = TABLE[0][0];
smallest=TABLE[0][0];
for (i=0;i<M;i++)
 for (j=0;j<N;j++)
  {
       if (TABLE[i][j] < smallest)
         smallest = TABLE[i][j];
       if (TABLE[i][j] > largest)
         largest = TABLE[i][j];
  }
cout<<"The value of smallest in TABLE is "<<smallest<<endl;
cout<<"The value of largest in TABLE is "<<largest<<endl;
```

23.

```
int FIND(int TABLE[][N], int ROW)
{
  int j, index = -1;
  for (j=0;j<N && index!= -1; j++)
     if (TABLE[ROW][j]==100)
          index = j;
  return index;
}
```

24.

```
int FIND(int TABLE[][N], int COL)
   {
     int i, index = -1;
     for (i=0;i<M && index!= -1; i++)
        if (TABLE[i][COL]==100)
              index = i;
     return index;
   }
```

25.

```
int YES_OR_NO(int TABLE[][N])
{
  int i,j;
  for (i=0;i<M;i++)
   for(j=0;j<n;j++)
     if (TABLE[i][j] == 100)
        return 1;
 return 0;
}
```