Part A - True or False Questions.

Question #1. Please answer **T** or **F** in the blanks to the left of each question.

___ a) In a C++ program, every }(not part of a comment) must be preceded by a matching {.

___ b) A function may call another function in C++.

___ c) In a syntactically correct C++ program the body of a *for* loop must be enclosed within braces { } .

___ d) In a syntactically correct C++ program the condition statement in an *if* statement or loop must be enclosed within parentheses ().

___ e) A function that returns a value might not have any parameters.

___ f) A *for* loop is best used if you don't know in advance how many iterations will be executed.

___ g) In executing a *do ... while* loop the body of the loop may never be executed at all.

___ h) A void function is one which has no parameters.

___ i) The usage ***a = f();*** implies that f cannot be a void function.

___ j) In a function a *return* with no expression can occur only in a void function.

PART B – Short Answer Questions – Answer all questions in the space provided.

Question 2. Will the following program terminate? if so, give the output:

```
int x = 100;
while(true)
{
   if ( x < 11 )
      break;
   x = x - 11;
}
cout << "x = " << x << endl;
```

a) Will the following program terminate? if so, give the output:

```
int x = 100;
while(true)
{
   if ( x < 11 )
      continue;
   x = x - 11;
}
cout << "x = " << x << endl;
```

Question 3.

Indicate the values of *a*, *b*, *c* and *d* as requested after each instruction in the following program segment:

Instructions	**value of a**	**value of b**	**value of c**	**value of d**
int a = 5, b = -5, c; float d = 3.5;				
c = a – b;				
a = 7 – 2 * 11 / 2 + 1;				
a = (7 – 2) * 11 / (2 + 1);				
a = d + 0.75;				
d = d + 0.75;				
a += 10;				
b++;				
b = 5 % 5 + 5 / 5 + 5 *5;				
a = b = c + 100;				

Question 4 -

Use these object definitions to evaluate the following. Show all details.

```
bool  P  =  false ;
bool  Q  =  true ;
bool  R  =  true ;
```

a) P && (Q || R)

b) P || Q && !P || !Q

Question 5.
Write the following algebraic expression as a C++ expression using the least number of brackets. Assume *a* and *x* are float variables.

$$x = \frac{2a^3 - 1}{2x + 1}$$

Question 6.
Determine the output of the following code segment, if any.

```
int x = 4;
int y = 5;
if( x > 4)
if(y > 4)
{
int z = x + y;
cout << "z = " << z << endl;
}
else
cout << "x + y" <<  (x + y) << endl;
```

Question 7.

```
int x, y, sum = 0;
for (x = 1; x <= 3; x++)
  for(y = x;  y <= 3; y++)
   {
     cout << x << '^ ' << y << endl;
     sum += (x + y);
    }
cout << sum;
```

a) What will the above program segment display?

b) (2 points) Rewrite the above program segment using *while* loops instead of *for* loops.

Question 8.
Rewrite the following using a switch statement:

```
 if( x == 1)
   x = x + 5;
 else if (x == 4)
   x += 7;
 else if ( x == 5 || x == 9)
   x -= 3;
 else
   x *= 2 + 3;
```

Question 9.
Complete the following program segment so that it produces the output given to its right.

```
prod = ___________;

for ( __________; __________; i++)
{
        prod = _____________________;

        cout << i << _______

              <<__________ << endl;
}
```

```
2! = 2
3! = 6
4! = 24
5! = 120
6! = 720
7! = 5040
```

Question 10. Read the following program, and then write on the lines below what it outputs.

```
#include<iostream >
using namespace std;
void A(int k, int& r)
{
  r = r + 10 * k;
}

void B(int n, int& r)
{
  ++n;
   r = r + 10 * n;
}

int main()
{
    int n, p, q;
    p = 40; q = 15; n = 3; A(p,q);
    cout <<"A1 " << p << '  ' << q << '  ' << n << endl;
```

```
    p = 40; q = 15; n = 3; A(p,n);
    cout <<"A2 " << p << '  ' << q << '  ' << n << endl;
    p = 40; q = 15; n = 3; A(n,n);
    cout <<"A3 " << p << '  ' << q << '  ' << n << endl;
    p = 40; q = 15; n = 3; B(n,q);
    cout <<"B1 " << p << '  ' << q << '  ' << n << endl;
    p = 40; q = 15; n = 3; B(p,n);
    cout <<"B2 " << p << '  ' << q << '  ' << n << endl;
    return 0;
}
```

On the following lines, write down everything the program outputs. Put one item in each blank.

A1 _____ _____ _____

_____ _____ _____ _____

_____ _____ _____ _____

_____ _____ _____ _____

_____ _____ _____ _____

PART C – Longer Answer Questions – Answer all questions in the space provided.

Question 11.

a) Write a C++ function *spacesOut()* which given a string *str* as a parameter, will return the string *str* with all the spaces removed. For example the function call spacesOut ("After the sunset") will return the string "Afterthesunset". The original string is not modified.

b) Give the function prototype for the function *spacesOut()*.

c) Give a typical function call for the function *spacesOut()*.

Question 12.

a) Write the definition of a class called *Product*. A *Product* object should represent a product stocked in a supermarket, e.g. a 500 gram can of corn. It should contain the following information: a *code* for the product, the *name* of the product, the *cost* of the product, and the quantity of the product currently in *stock* Assume the code and the name are represented by strings of characters. Include the following constructor and member functions in the class definition.

- a constructor *Product(string code, string name)* which creates a new product with the given code and name. Initially, the cost and the quantity of the product should be set to zero.

- a member function *getName ()* with no arguments, that returns the name of the product.

- a member function *addStock(int n)* that will add *n* to the quantity of the product in *stock*. If *n* is negative, the method should display an error message and should not modify the quantity of the product in *stock*.

- A member function *subStock(int n)* that will subtract *n* from the quantity of the product in *stock*. If *n* is negative or *n* is larger than the available quantity of the product, the method should display an appropriate error message and should not modify the current quantity of the product in *stock*.

- A member function *outOfStock()*with no arguments, that will return the value true if there is none of this product in *stock*; otherwise it will return false.

- A member function *setCost(double cost)* that will set the cost of the product to *cost* **.** If *cost* is negative, the method should display an error message and should not modify the cost of the product in stock.

- A member function *getCost()*with no arguments that will return the cost of the product.

It should not be possible to manipulate the fields of a *Product* object except by using the methods listed above.

b) Implement the member function *subStock(int n)*.

c) Implement the member function *outOfStock()*.

d) One of the items in the supermarket is a box of *Raisin Bran Cereal* whose code is *RBL987*. Write the C++ statement(s) to declare a *Product* object, *RaisinBran*, which represents this product.

e) Use a class member function to output the cost of the object *RaisinBran.*.

PART D – Complete Program

Question 13.
Write a complete program to read in a collection of exam grades ranging in value from 0 to 100. Your program should display the category of each score. It should also count and display the number of scores in each category.
The categories are:

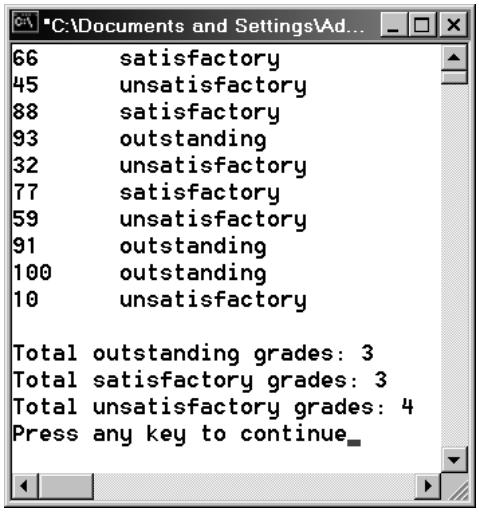
Outstanding: grades 90 to 100
Satisfactory: grades 60 to 89
Unsatisfactory: grades 0 to 59.

The exam grades are to be read from the file *question13.dat* which already exists. You don't know how many grades are in the file, but you know that they are all valid grades.

Sample data file:

```
66  45  88  93  32  77  59  91  100  10
```

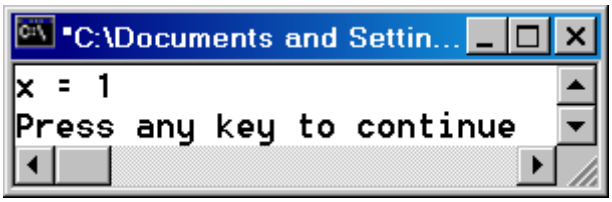
Sample output to the screen:



Comp 218
Final Examination - SOLUTION

PART A – Question 1
a) T b) T c) F d) T e) T f) F g) F h)F i) t j) T

PART B
Question 2.
a)



b) No Output (Infinite loop)

Question 3.

Instructions	value of a	value of b	value of c	value of d
int a = 5, b = -5, c; float d = 3.5;	5	-5	?	3.5
c = a – b;	5	-5	10	
a = 4 – 2 * 5 / 2 + 1;	-3			
a = (4 – 2) * 5 / (2 + 1);	18			
a = d + 0.75;	4			
d = d + 0.75;				4.25
a += 100;	14			
b++;		-4		
b = 5 % 5 + 5 / 5 + 5 *5;		26		
a = b = c + 10;	110	110	10	

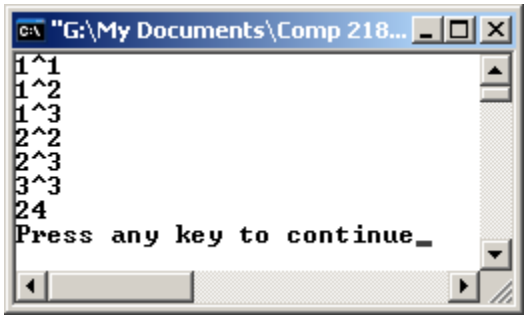
Question 4
a) P&& (Q|| R) → F && (T || T) → F && (T) → F
b) P || Q && !P || !Q → F || T && !F || ! T → F || T && T || F
→ F || T || F → T || F → T

Question 5.
x = (2* a *a * a – 1) / (2 * x + 1) ;

Question 6
No output

Question 7

a)



b)
```
int x, y, sum = 0;
x = 1;
while( x <= 3)
{
  y = x;
  while ( y <=3)
  {
    cout << x << '^ ' << y << endl;
    sum += (x+y);
    y++;
  }
  x++;
}
cout << sum;
```

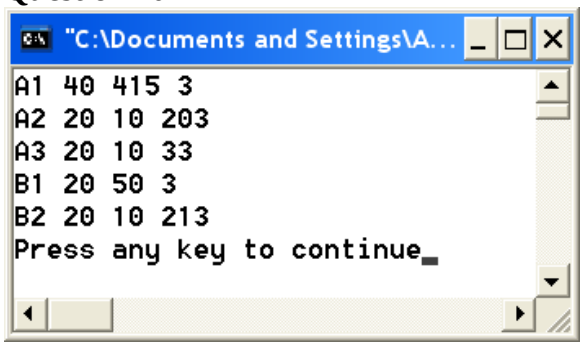
Question 8

a)
```
switch(x)
  {
     case 1: x += 5;
             break;
     case 4: x+=7;
              break;
     case 5: case 9:
             x -= 3;
             break;
     default: x*= 2 + 3;
  };
```

Question 9

```
prod = 1;
for ( int i = 2 : i <= 7; i++)
{
        prod = prod * i;
        cout << i << " ! = "
             <<  prod << endl;
}
```

Question 10



PART C

Question 11

a)
```
string spacesOut(string str)
{
   string temp;
   for (int j = 1; j<= str.length(); j++)
     if(str.substr(j-1,1) <> " ") temp+=(str.substr(j-1,1);
   return temp
 }
```

b) string spacesOut (string str); or string spacesOut (string);

c) cout << "The modified string is " << spacesOut (str) << endl;

Question 12

a)
```
class Product
{
  private:
        string code;
        string name;
        float cost;
        int stock;
public:
        Product(string code, string name);
        string getName();
        void addStock(int n);
        void subStock(int n);
        bool outOfStock();
        void setCost(double cost);
        float getCost();
};
```

b) void Product::subStock(int n)
```
   {
     if (n < 0 || n > stock)
          cout << “invalid request\n”;
     else
          stock -=n;
    }
```
c) bool Product::outOfStock()
```
   { return stock <= 0;}
```
d) Product RaisinBran(“Raisin Bran Cereal”, “RBL9873”);

e) cout << “Cost of cereal is $“ << RaisinBran.getCost()<<endl;

PART D

Question 13

```
#include <iostream>
#include <fstream>
using namespace std;
int main()
{
  int sat = 0, unsat = 0, outst = 0;
  int grade;
  ifstream fin;
  fin.open("q13.dat");
  fin >> grade;
  while (fin)
  {
    if (grade >= 90)
        {
            cout << grade << "\toutstanding\n";
            outst++;
        }
        else if (grade >= 60)
        {
            cout << grade << "\tsatisfactory\n";
            sat++;
        }
        else
        {
            cout << grade << "\tunsatisfactory\n";
            unsat++;
        }
        fin >> grade;
    }
    fin.close();
    cout << "Total outstanding grades: " << outst << endl;
    cout << "Total satisfactory grades: " << sat << endl;
    cout << "Total unsatisfactory grades: " << unsat << endl;
    return 0;
}
```