Midterm Exam Solution

Question 1: (25 pts) Multiple choice questions.

1. What is the output of following code?

```
int i = 1, Total = 10;
while ( (i <= 5) || (Total > 0)  ) {
  cout<< Total<<"  ";
  if(Total == 0)
     break;
  i++;
  Total = Total - i;
}
```

(a) 10 8 5 1
(b) 10 8 5 1 -4
(c) 10 8 6 4 2
(d) 10 9 8 7 6
(e) 10 9 8 7 6 5 4 3 2 1

Correct Answer: **b**

2. What is the output of following code?

```
int num = 0;
while(!false){
        if(num == 10)
                cout<<"Hello World";
                break;
                num++;
}
```

(a) Nothing will be printed on the output screen
(b) Hello World will be printed 9 times
(c) Hello World will be printed 10 times
(d) Hello World will be printed 11 times
(e) None of the above

Correct Answer: **a**

3. What is the output of following code?

```
int Rank = 2;
switch(Rank){
      case 1:
      cout<<"Rank 1"<<endl;
case 2:
      cout<<"Rank 2"<<endl;
case 3:
      cout<<"Rank 3"<<endl;
default:
      cout<<"Invalid Rank !"<<endl;
 }
```

(a) Rank 2

(b) Rank 2
Rank 3

(c) Rank 2
Rank 3
Invalid Rank !

(d) Invalid Rank !

(e) Syntax Error

Correct Answer: **c**

4. What is the output of following code?

```
int A = 1;
char Letter = A;
switch(Letter){
      case 'A':
      cout<<"Letter A"<<endl;
     break;
      case '1':
      cout<<"Letter 1"<<endl;
     break;
      case 'a':
      cout<<"Letter "<<Letter<<endl;
     break;
      default:
      cout<<"Invalid Letter!\n";
      }
```

(a) Letter A
(b) Letter 1
(c) Letter a
(d) Invalid Letter !
(e) Syntax Error

Correct Answer: **d**

5. How many times the following loop run?

```
for(int i = 0, j = 0; i + j < 20;)
{
    i += 4;
    j += 2;
}
```

(a) 2
(b) 4
(c) 5
(d) infinite

Correct Answer: **c**

Question 2: (25 pts) Write the output of following code snippets.

a)
`int main()`
`{`
`  int num=10, i=0;`
`  while(i < num)`
`  {`
`  if(i % 2 == 0)`
	0-9-1-8-2-7-3-6-4-5-

{
cout<<i/2<< "-";
}
else
{
cout<<num-i/2-1<< "-";
}
i++;
}
return 0;
}	

b)
int main()
{
int counter = 0;
counter -= 2;
while(counter != 10)
{
cout << counter << " ";
counter += 2;
}
return 0;
}
	-2 0 2 4 6 8

c)
int main()
{
for(int x=1; x<=3; x++)
{
for(int y=1; y<=3; y++)
{
if(x == y)
{
cout<<x+y<<endl;
}
}
}
return 0;
}
	2
	4
	6

d)
int main()
{
int x, y, z=4;
x += 2;
y -= 1;
z *= 3;
cout <<x<< " " <<y<< " " << z;
return 0;
}
	Error/ Uninitialized x and y

e)
int main()
{
int x = 0;
for(int i=1; i<10; i*=2)
{
x++;
cout << x << endl;
}
cout << x << endl;
	1
	2
	3
	4
	4

`    return 0;` `}`	

Question 03: (05 pts) Find errors in the following code.

```
// This program averages 3 test scores.
// It uses the variable perfectScore as a flag.
include <iostream>
using namespace std;

int main()
{
   cout << "Enter your 3 test scores and I will ";
        << "average them:";
   int score1, score2, score3,
   cin >> score1 >> score2 >> score3;
   double average;
   average = (score1 + score2 + score3) / 3.0;
   if (average = 100);
       perfectScore = true;  // Set the flag variable
   cout << "Your average is " << average << endl;
   bool perfectScore;
   if (perfectScore);
   {
      cout << "Congratulations!\n";
      cout << "That's a perfect score.\n";
      cout << "You deserve a pat on the back!\n";
      return 0;
}
```

1. `include<iostream>` // # is missing
2. `if (average = 100);` // == must be used when using condition and semicolon mean condition is always true.
3. `perfectScore` // Uninitialized variable

Question 04: (15 pts) Running on a particular trade mill, you burnt 3.6 calories per minute. Write a program in C++ that uses a loop to display the number of calories burned after 5, 10, 15, 20, 25 and 30 minutes.

```
#include <iostream>
using namespace std;
int main()
{
    const float CALORIES_BURNED_PER_MIN = 3.6;

    float calories_burned = 0;

    for (int i = 5; i <= 30; i += 5)
    {
        calories_burned = (i * CALORIES_BURNED_PER_MIN);

        cout << "Calories burned after " << i
            << " minutes = "
            << calories_burned
            << endl;
    }

    return 0;
}
```

Question 05: (15 pts) Write a program that uses nested loops to collect data and calculate the average rainfall over a period of years. The program should first ask for the number of years. The outer loop will iterate once for each year. The inner loop will iterate twelve times, once for each month. Each iteration of the inner loop will ask the user for the inches of rainfall for that month. After all iterations, the program should display the number of months, the total inches of rainfall, and the average rainfall per month for the entire period.
Input Validation: Do not accept a number less than 1 for the number of years. Do not accept negative numbers for the monthly rainfall.

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

int main()
{
    const int NUM_OF_MONTHS = 12;
    int num_of_years,
        total_num_of_months = 0;

    double rainfall,
        total_rainfall=0,
        average_rainfall_per_month;

    cout << "Enter number of years: ";
    cin >> num_of_years;
    cout << "You've entered "
        << num_of_years
        << " years."
        << endl;

    for (int i = 0; i < num_of_years; i++)
    {
        cout << "Year " << (i + 1) << ": " << endl;
        for (int j = 0; j < NUM_OF_MONTHS; j++)
        {
            cout << "How much rainfall (inches)\n"
                << "for month " << (j + 1) << ": ";
            cin >> rainfall;

            total_rainfall += rainfall;

            total_num_of_months++;
        }
    }

    average_rainfall_per_month = total_rainfall /
        total_num_of_months;

    cout << setprecision(2) << fixed;
    cout << "Total number of months = "
        << total_num_of_months << endl;

    cout << "Total inches of rainfall = "
        << total_rainfall << endl;

    cout << "Average rainfall per month = "
        << average_rainfall_per_month << endl;


    return 0;
}
```

Question 06: (15 pts) Write a program that will predict the size of a population of organisms. The program should ask the user for the starting number of organisms, their average daily population

increase (as a percentage), and the number of days they will multiply. A loop should display the size of the population for each day.
Input Validation: Do not accept a number less than 2 for the starting size of the population. Do not accept a negative number for average daily population increase. Do not accept a number less than 1 for the number of days they will multiply.

```
#include <iostream>
using namespace std;
int main()
{
    float starting_num_of_organisms,
        average_daily_population_increase,
        size_of_daily_population;

    int num_of_days_to_multiply;

    cout << "Enter the starting number of organisms: ";

    while (!(cin >> starting_num_of_organisms) ||
        starting_num_of_organisms < 2)
    {
        cout << "Oops. Starting number must be 2 or greater." << endl;
        cout << "Enter the starting number of organisms: ";
        cin.clear();
        cin.ignore(123, '\n');
    }

    cout << "Enter average daily population increase (%): ";

    while (!(cin >> average_daily_population_increase) ||
        average_daily_population_increase < 0)
    {
        cout << "Oops. Average daily population increase\n"
            << " must be greater than 0. Enter average\n"
            << " daily population increase (%): ";
        cin.clear();
        cin.ignore(123, '\n');
    }
    average_daily_population_increase *= .01;

    cout << "Enter number of days they will multiply: ";

    while (!(cin >> num_of_days_to_multiply) ||
        num_of_days_to_multiply < 1)
    {
        cout << "Oops. Number of days must NOT be less\n"
            << "than 1. Enter number of day they will\n"
            << "multiply: ";
        cin.clear();
        cin.ignore(123, '\n');
    }

    for (int i = 0; i < num_of_days_to_multiply; i++)
    {
        cout << "Poplation size for day " << (i + 1);
        cout << ": " << starting_num_of_organisms
            << endl;

        starting_num_of_organisms +=
            (starting_num_of_organisms *
                average_daily_population_increase);
    }

    return 0;
}
```