Home Work No.7

Content:

1) Selection Structure (Making Decisions) and Random Number Generation
2) Output Problems
3) Repetition Structure Practice Codes

……

Run following programs in separate .cpp files and carefully understand the output.

```
/......................Switch with ranges ............//
//……………………………1.cpp……………………..//
int main()

{

        int x;

        cout<<"\nEnter Value between ranges \n\t 1 to 4\n\t 5 to 7\n\t 8 to 10";
        cin>>x;


        switch(x)
        {
                case 1 ... 4:

                        cout<<"1 . . . 4";
                        break;

                case 5 ... 7:
                        cout<<"5 . . . 7";
                        break;

                case 8 ... 10:
                        cout<<"8 . . . 10";
                        break;

                default:
```

```
                              cout<<"wrong input";

            }

return 0;

}
```
……………………………………………..2.cpp…………………………………………………………………

//Execute the following code. Use of **Pseudo Random number generator** without seed. Convert code using Switch statement

```
#include <iostream>
#include <cstdlib>
using namespace std;
int main()
{
            int day;
            day = rand();
            cout<<"\nValue generated from random Function";
            cout<<endl<<day<<endl;

            cout<<"Normalizing Day 1 to 7 according to our problem";
            day = (day % 7) + 1;
            cout<<endl<<day<<endl;

            if (day == 1)
                        cout << "Sunday\n";
            else if (day == 2)
                        cout << "Monday\n";
            else if (day == 3)
                        cout << "Tuesday\n";
           else if (day == 4)
                        cout << "Wednesday\n";
            else if (day == 5)
                        cout << "Thursday\n";
            else if (day == 6)
                        cout << "Friday\n";
            else if (day == 7)
                        cout << "Saturday\n";
            ///Remember no need of default case now
            return 0;
}
```

…………………………………………………………….3.cpp……………………………………………………………………………….
//Execute the following code. Use of **Pseudo Random number generator** with seed. Convert code using Switch statement

```
#include <iostream>
#include <cstdlib>

using namespace std;
int main()
{

        int day, seed;
        cout<<"Enter value of seed: ";
        cin>>seed;//Taking seed as input from user

        srand(seed);//function to incorporate seed
        day = rand();

        cout<<"\nValue generated from random Function";
        cout<<endl<<day<<endl;


        cout<<"Normalizing Day 1 to 7 according to our problem";
        day = (day % 7) + 1;
        cout<<endl<<day<<endl;

        if (day == 1)
                cout << "Sunday\n";
        else if (day == 2)
                cout << "Monday\n";
        else if (day == 3)
                cout << "Tuesday\n";
       else if (day == 4)
                cout << "Wednesday\n";
        else if (day == 5)
                cout << "Thursday\n";
        else if (day == 6)
                cout << "Friday\n";
        else if (day == 7)
                cout << "Saturday\n";
        ///Remember no need of default case now

return 0;
}
```

```
//…………………………………………………………….4.cpp……………………………………………………………………….
//Execute the following code. Use of Random number generator with seed taken as clock time of
system. Convert code using Switch statement

#include <iostream>
#include <cstdlib>
#include <ctime>

using namespace std;
int main()
{

        int day, seed;
        cout<<"Enter value of seed: ";
        srand(time(NULL));//Taking seed as input from current time (number of seconds elapsed in your
machine)
        day = rand();

        cout<<"\nValue generated from random Function";
        cout<<endl<<day<<endl;


        cout<<"Normalizing Day 1 to 7 according to our problem";
        day = (day % 7) + 1;
        cout<<endl<<day<<endl;

        if (day == 1)
                cout << "Sunday\n";
        else if (day == 2)
                cout << "Monday\n";
        else if (day == 3)
                cout << "Tuesday\n";
       else if (day == 4)
                cout << "Wednesday\n";
        else if (day == 5)
                cout << "Thursday\n";
        else if (day == 6)
                cout << "Friday\n";
        else if (day == 7)
                cout << "Saturday\n";
        ///Remember no need of default case now

return 0;
}
```

###

…………………………………………………………………. Article on …………………………………………………………………………..

…………………………………………………. Random Number Generation……..……………………………………………………

Random Number Generation C++ It is often useful to generate random numbers to produce simulations or games (or homework problems :) One way to generate these numbers in C++ is to use the function rand(). Rand is defined as:

```
#include<cstdlib>
int rand();
```

The rand function takes no arguments and returns an integer that is a pseudo-random number between 0 and RAND_MAX. On transformer, RAND_MAX is 2147483647.

Which in C++ is given by formula:

$$(1103515245\ S_i + 12345) \bmod 2^{31}$$

What is a pseudo-random number? It is a number that is not truly random, but appears random. That is, every number between 0 and RAND_MAX has an equal chance (or probability) of being chosen each time rand() is called. (In reality, this is not the case, but it is close). For example, the following program might print

```
cout <<rand()<< endl;
 cout << and()<< endl;
 cout << and()<< endl;
```

```
1804289383
 846930886
1681692777
```

Here we "randomly" were given numbers between 0 and RAND_MAX. What if we only wanted numbers between 1 and 10? We will need to scale the results. To do this we can use the modulus (%) operator. Any integer modulus 10 will produce a number from 0-9. In other words, if we divide a number by 10, the remainder has to be from 0 to 9. It's impossible to divide a number by 10 and end up with a remainder bigger than or equal to ten.

In our case:

```
1804289383% 10 = 3
846930886% 10 = 6
1681692777% 10 = 7
```

Consequently, we can take rand() % 10 to give us numbers from 0-9. If we want numbers from 1-10 we can now just scale up by adding one. The final result is:

```
cout << (rand() % 10) + 1 << endl;
```

If you run this program many times, you'll quickly notice that you end up with the same sequence of random numbers each time. This is because these numbers are not truly random, but pseudorandom. Repeat calls to rand merely return numbers from some sequence of numbers that appears to be random. Each time we call rand, we get the next number in the sequence.

If we want to get a different sequence of numbers for each execution, we need to go through a process of randomizing. Randomizing is "seeding" the random number sequence, so we start in a different place. The function that does this is srand() which takes an integer as the seed:

```
void srand(int seed);
```

It is important to only invoke the srand call ONCE at the beginning of the program. There is no need for repeat calls to seed the random number generator (in fact, it will make your number less evenly distributed).

A commonly used technique is to seed the random number generator using the clock. The time() function will return the computer's time. On transformer, this is expressed in terms of the number of seconds that have elapsed since Jan 1, 1970 (the Epoch). The function time(NULL) will return the number of seconds elapsed in computer time.

```
#include <ctime>
srand(time(NULL));
cout << (rand() %10) + 1;
```

This produces different values each time the program is run.

Reference: [***http://www.math.uaa.alaska.edu/~afkjm/csce211/handouts/RandomFunction***](http://www.math.uaa.alaska.edu/~afkjm/csce211/handouts/RandomFunction)

Output Problems:
What will be the output of following codes:

```
int main()
{
	int choice = 5;
	switch (choice) {
	default:
		cout << "\nI am in Default";
	case 1:
		cout << "\nI am in case 1";
		break;
	case 2:
		cout << "\nI am in case 2";
		break;
	case 3:
		cout << "\nI am in case 3";
		break;
	}
return 0;
}
```

Output:

```
int main()
{
	int choice = 2;
	switch (choice) {
	default:
		cout << "\nI am in Default";
	case 1:
		cout << "\nI am in case 1";
	case 2:
		cout << "\nI am in case 2";
	case 3:
		cout << "\nI am in case 3";
	}
return 0;
}
```

Output:

```
int main()
{
	float num1 = 1.1;
	double num2 = 1.1;
	if (num1 == num2)
		cout << "Stanford";
	else
		cout << "Islamabad";

return 0;
}
```

Output:

```
int main()
{
	int y = 0;
```

Output:

```
	switch (y){

	case 0:  y = y + 5;

	case 1:  y = y / 2;
	case 2:  y = y * 3;
	case 3:  y = y + 10;

	default: y = y%3;

	}
	cout << y << endl;
return 0;
}
```

```
int main()
{
	int i = 3, j = 3, k = 3;
	if (--i - 7 && j++ < ++k)
		cout<< ++i;
	else
		cout<< i<< j<< k;
return 0;
}
```

Output:

```
int main()
{

int i = j = k = 3;
i -= j -= k;
cout<< i<< j<< k;
return 0;
}
```

Output:

```
int main()
{

int y = 0, x = 1;
if (y != 0);
{
	if (x != 0);
		result = x / y;
}
else {
	cout<< Error: y is equal to zero\n ;
}
return 0;
}
```

Output:

Dry Run following programs and carefully understand the output.

…………code Segment 1 ……………………………..

```
const int MIN_NUMBER = 1, MAX_NUMBER = 10;

        int num = MIN_NUMBER;


   while (num <= MAX_NUMBER)
   {
        cout << num << setw(10) << (num * num) << endl;
        num++;
        }
```

…………code Segment 2 ……………………………..

```
int i=10;
while (i>0)
{
        cout<<setw(i)<<"*"<<endl;
        i--;
}
```

…………code Segment 3 ……………………………..

```
int i=5;

while (i>0)
{
        cout<<setw(i)<<"*"<<endl;
        i--;
}
while (i<4)
{
        cout<<setw(i+2)<<"*"<<endl;
        i++;
}
```

…………code Segment 4 ……………………………..

```
int a = 0;
        while (a < 100) {
        if (a % 5 == 0)
        cout << "*"<<endl;
        a++;
        }
        cout << '\n';
```

…………code Segment 5 ……………………………..

```
int main() {


        int x = 2;
        int y = 10;

        cout <<++x * y--;

        cout <<x++ * y--<<" "<<--x * --y <<" "<<++x * ++y<<endl;


return 0;
}
```

…………code Segment 6……………………………..

```
int main() {


        int x = 1;

        while (x <= 50)
        {
                if (x % 3 == 0 || x % 5 == 0)
                        cout << x << " ";
                x = x + 1;

        }

return 0;
}
```

What is wrong with the following code Segments? Explain.

…………code Segment 1 ……………………………..

```
int num1 = 0, num2 = 10, result;
num1++;
result = ++(num1 + num2);
cout << num1 << " " << num2 << " " << result;
```

…………code Segment 2 ……………………………..

```
int num1 = 0;

while (num1<=10)
```

```
          cout<<num1;
          num1++;
```

…………code Segment 3 ……………………………..

```
int num = 1;

while ( )
{
          cout<<num;
          num1++;
}
```

…………code Segment 4 ……………………………..

```
int num, bigNum, power, count;
cout << "Enter an integer: ";
cin >> num;
cout << "What power do you want it raised to? ";
cin >> power;
bigNum = num;
while (count++ < power);
          bigNum *= num;

cout << "The result is "<< bigNum << endl;
```

…………code Segment 5 ……………………………..

```
int count = 1, total;
while (count <= 100)
          total += count;
cout << "The sum of the numbers 1-100 is ";
cout << total << endl;
```