**IT 1050 – Programming Logic**  
Midterm Test 1

Dhruvil Patel

*30 points (extra credit problem +2)*

1. (4 points) Identify and briefly describe 3 types of control structures that we can use to order the statements in our programs?

Ans. The three types of control structures are selection statements, iteration statements and jump statements. Selection statements enable us to branch to different sections of code. Iteration statements consist of loops that perform a specific task repeatedly until the given condition is met. Jump statements transfer control to different sections of the code.

1. (4 points) Create an infinite while loop. Use a Boolean variable called keepLooping that set to true in the loop’s termination condition. Hint: Use CTRL+C or Debug -> Terminate All to end the program.

Ans. Code:

int i = 12;

Boolean keepLooping = true;

while(keepLooping)

{

if (i <= 11)

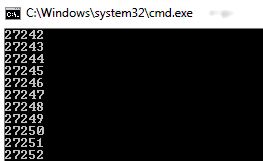
keepLooping = false;

i++;

Console.WriteLine(i);

}

Execution:



1. (4 points) Write a while loop to prints 2 through 128 in brackets, each on a new line. You should initialize your loop control variable to 2. Output the value of the loop control variable each time through the loop. Use a condition that ends the loop after 128 is printed.

[2]

[4]

[8]

[16]

[32]

[64]

[128]

Ans. Code:

int i = 2;

while (i <= 128)

{

{

if ((i%2) == 0)

{

Console.WriteLine("[{0}]", i);

}

else if ((i%2) != 0)

{

Console.Write("");

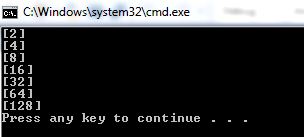
}

i\*=2;

}

}

Execution:



1. (4 points) Write a for loop that prints 49 through 1 separated by a comma. Note, you will need to use a condition inside of the loops so it does not print the comma the last time through (no newlines).

49, 48, 47, …, 3, 2, 1

Ans. Code:

string j = ",";

int i;

for(i=49; i>=1; --i)

{

Console.Write(i);

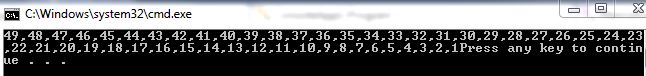
if(i>=2)

{

Console.Write(j);

}

}  
Execution:



1. (4 points) Write a while loop that prints all odd numbers 1 through 21 separated by spaces (no newlines).

1 3 5 7 9 11 13 15 17 19 21

Ans. Code:

int i = 1;

while(i<=21)

{

{

if((i%2)==0)

{

Console.Write(" ");

}

else if ((i%2)!=0)

{

Console.Write(i);

}

i++;

}

}

Execution:



1. (5 points) What is the output for the following code?

int n = 8;

int i = 10; // initialize

do {

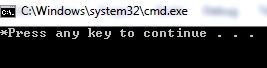
Console.Write("\*");

i++; // update!

} while (i < n); // test condition

Ans. The output of the following code is a “\*” symbol.

Execution:



What would the output be with the exact same code but using a while statement as opposed to a do-while?

Ans. Code:

int n = 8;

int i = 10;

while(i<n)

{

Console.Write("\*");

i++;

}

Execution:



The code would not show any output as in a while loop, the condition is checked first and if it is true, only then the loop is executed. As opposed to this, in a do-while loop, the loop is executed at least once before checking the condition.

1. (5 points) Explain how do we combine multiple Boolean values? Write an if statement that outputs “Let’s go outside!” when both Boolean values are false.

boolean icyRain;  
boolean tornadoWarning;

Ans. Boolean values can be combined using logical operators.

Code:

bool icyRain = false, tornadoWarning = false;

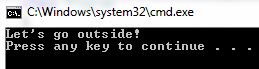
if(icyRain==false&&tornadoWarning==false)

{

Console.WriteLine("Let's go outside!");

}

Execution:



1. Extra Credit: Use nested loops to print the following to the console:

123454321

1234321

12321

121

1

Ans.

Code:

int numberoflayer = 5, Space, Number;

for(int i=1; i<=numberoflayer; i++)

{

for (Space = 1; Space <= (numberoflayer - i); Space++)

Console.Write("");

for (Number = 1; Number <= i; Number++)

Console.Write(Number);

for (Number = (i - 1); Number >= 1; Number--)

Console.Write(Number);

Console.WriteLine();

}

Execution:

