**ASSIGNMENT # 1**



**Spring 2019**

**CSE102 Computer Programming**

Submitted by: **SHAH RAZA**

Registration No. : **18PWCSE1658**

Class Section: **B**

“On my honor, as student of University of Engineering and Technology, I have neither given nor received unauthorized assistance on this academic work.”

Student Signature: \_\_\_\_\_\_\_\_\_\_\_\_\_\_

Submitted to:

**Engr. Madiha Sher**

March 4, 2019

Department of Computer Systems Engineering

University of Engineering and Technology, Peshawar

1. Any character is entered through the keyboard; write a program to determine whether the character entered is a capital letter, a small case letter, a digit, or a special symbol.

The following table shows the range of ASCII values for various characters.

**Characters ASCII values**

A – Z 65 – 90

a – z 97 – 122

0 – 9 48 – 57

Special symbols 0 – 47, 58 – 64, 91 – 96, 123 – 127

**Code:**

#include <iostream> /\* library for

writing the output to console window\*/

using namespace std;

int main()

{

//Variable declaration

char C;

//Ask for the value of C from user

cout << "Enter the Character: ";

cin>>C;//Value of C

//One line space

cout<<"\n";

//Condition for Capital letters

if (C>=65 && C<=90)

cout<<"The character you entered is a Capital letter";

//Condition for lowercase letters

else if(C>=97 && C<=122)

cout<<"The character you entered is a lowercase letter";

//Condition for Digits

else if(C>=48 && C<=57)

cout<<"The character you entered is a Digit";

else

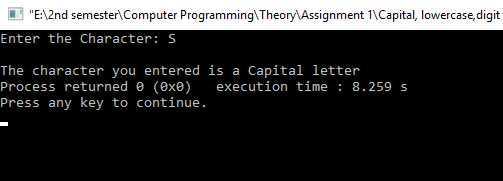
cout<<"The character you entered is a Special character";

return 0;

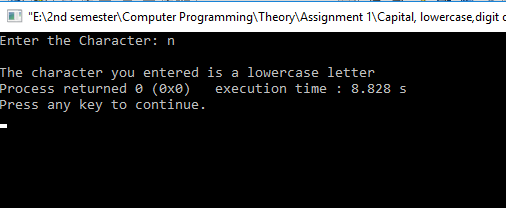
}

### **Output (Compilation, Debugging & Testing):**

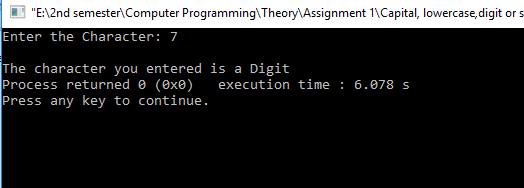
**Capital Letter:**

****

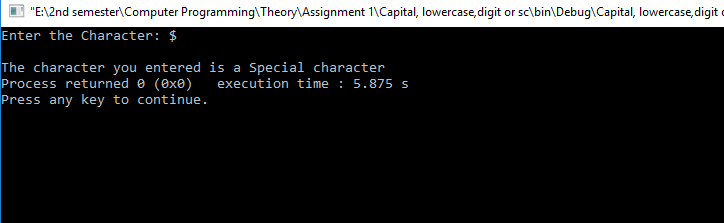
**Lowercase Letter:**

****

**Digit:**

****

**Special character:**

****

1. The following table contains earthquake magnitude ranges on the Richter scale and their descriptors:

|  |  |
| --- | --- |
| **Magnitude** | **Descriptor** |
| Less than 2.0 | Micro |
| 2.0 to less than 4.0 | Minor |
| 4.0 to less than 5.0 | Light |
| 5.0 to less than 6.0 | Moderate |
| 6.0 to less than 8.0 | Major |
| 8.0 to less than 10.0 | Great |
| 10.0 or more | Meteoric |

Write a program that reads a magnitude from the user and displays the appropriate descriptor as part of a meaningful message. For example, if the user enters 5.5 then your program should indicate that a magnitude 5.5 earthquake is considered to be a moderate earthquake.

**Code:**

#include <iostream> /\* library for

writing the output to console window\*/

using namespace std;

int main()

{

//Variable declaration

float num;

cout << "Enter the magnitude of Earth-quake: ";

//Input the value of num

cin>>num;

//One line space

cout<<"\n";

//Condition for Micro earthquake

if(num<2)

cout<<"A magnitude "<< num <<" earthquake is considered to be a Micro earthquake.";

//Condition for Minor earthquake

else if(num<4)

cout<<"A magnitude "<< num <<" earthquake is considered to be a Minor earthquake.";

//Condition for Light earthquake

else if (num<5)

cout<<"A magnitude "<< num <<" earthquake is considered to be a Light earthquake.";

//Condition for Moderate earthquake

else if (num<6)

cout<<"A magnitude "<< num <<" earthquake is considered to be a Moderate earthquake.";

//Condition for Major earthquake

else if (num<8)

cout<<"A magnitude "<< num <<" earthquake is considered to be a Major earthquake.";

//Condition for Great earthquake

else if (num<10)

cout<<"A magnitude "<< num <<" earthquake is considered to be a Great earthquake.";

//Condition for Meteoric earthquake

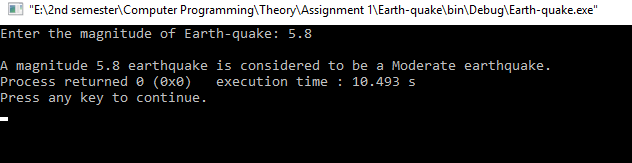
else

cout<<"A magnitude "<< num <<" earthquake is considered to be a Meteoric earthquake.";

return 0;

}

### **Output (Compilation, Debugging & Testing):**



1. Read a 3 digit number N from keyboard and find individual digits in unit’s place (U), ten’s place (T) and hundred’s place (H). Check U + T\*10 + H\*102 = N (given no)

**Code:**

#include <iostream> /\* library for

writing the output to console window\*/

using namespace std;

int main()

{

//Variable declaration

int N,U,T,H;

cout << "Enter a 3-digit Number: ";

//Input value of N

cin>>N;

//One line space

cout<<"\n";

//Formula for Unit's Place

U=N%10;

//Formula for Ten's Place

T=(N%100)/10;

//Formula for Hundred's Place

H=N/100;

//Display unit's Place

cout<<”Unit's place: “<<U<<endl;

//Display Ten's Place

cout<<”Ten's place: "<<T<<endl;

//Display Hundred's Place

cout<<”Hundred's place: "<<H<<endl<<endl;

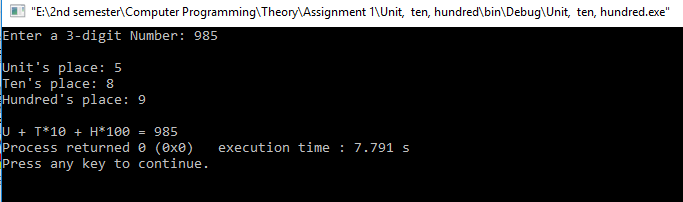
//Varification of U+T\*10+H\*100=N

cout<<”U+T\*10+H\*100= “<<U+T\*10+H\*100;

return 0;

}

### **Output (Compilation, Debugging & Testing):**



1. Write a RANDOM NUMBER GUESSING GAME –
   * Hard code a random number as answer
   * Ask user for input of Number
   * If INPUT is greater, print **"Try again with a LOWER number"**
   * If INPUT is lower, print **"Try again with a HIGHER number"**
   * If INPUT is equal, print **">>>> YOU WIN....! <<<<"**
   * Give the user 3 tries to guess the number

**Code:**

#include <iostream> /\* library for

writing the output to console window\*/

using namespace std;

int main()

{

//Generate a random number

int num=rand();

//Variable declaration

int guess;

cout << "Guess the Number: ";

//This loop runs 3 times

for(int x=1;x<=3;x++)

{

//Input value of guess

cin>>guess;

//when guess is greater than num

if (guess>num)

{

//Message displayed when user guess wrong 3 times

if (x==3)

cout<<"Game Over!!!";

//Message displayed when user guess is greater than num

else

cout<<"Try again with a LOWER number: ";

}

//when guess is less than num

else if (guess<num)

{

//Message displayed when user guess wrong 3 times

if (x==3)

cout<<"Game Over!!!";

//Message displayed when user guess is less than num

else

cout<<"Try again with a HIGHER number: ";

}

//When guess is equal to num

else

cout<<">>>>YOU WIN....!<<<<"<<endl;

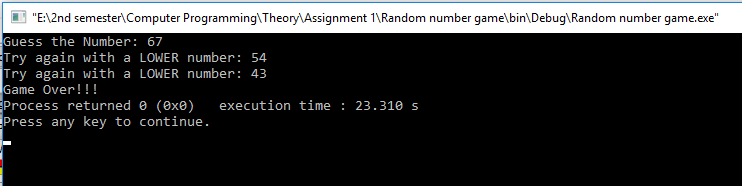
}

return 0;

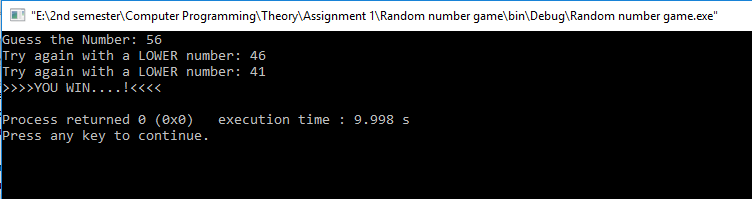
}

### **Output (Compilation, Debugging & Testing):**

**Failed Attempt:**

****

**Successful Attempt:**

****

1. Check whether a year is leap year or not?

In the calendar three criteria must be taken into account to identify leap years:

* + The year can be evenly divided by 4;
  + If the year can be evenly divided by 100, it is NOT a leap year, unless;
  + The year is also evenly divisible by 400. Then it is a leap year.

**Code:**

#include <iostream> /\* library for

writing the output to console window\*/

using namespace std;

int main()

{

//Variable declaration

int Year;

cout << "Enter Year: ";

//Input value of Year

cin>>Year;

cout<<"\n";

//Condition for leap year

if(Year%4==0 && (Year%100!=0 || (Year%100==0 && Year%400==0)))

{

cout<<"This is a leap year";

}

//Not a leap year

else

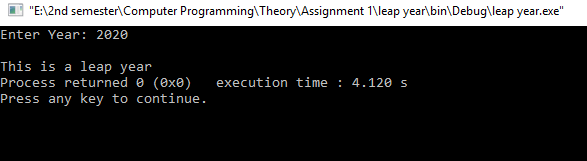
cout<<"This is a not leap year";

return 0;

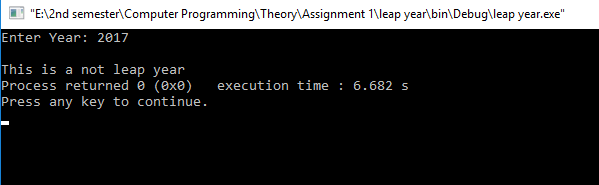
}

### **Output (Compilation, Debugging & Testing):**

**Leap Year:**



**Not A Leap Year:**



1. The total distance traveled by vehicle in ‘t’ seconds is given by distance=ut+1/2at2 where ‘u’ and ‘a’ are the initial velocity (m/sec) and acceleration (m/sec2). Write a C++ program to find the distance traveled at regular intervals of time given values of ‘u’ and ‘a’. The program should provide the flexibility to the user to select his own time intervals.

**Code:**

#include <iostream> /\* library for

writing the output to console window\*/

using namespace std;

int main()

{

//Variable declaration

int t1,t2;

float u,a;

cout << "Initial Velocity (m/s): ";

//Input value of u

cin>>u;

cout<<"\nAcceleration (m/s^2): ";

//Input value of a

cin>>a;

cout<<"\nTime interval: \n";

cout<<"t1 (s): ";

//Input value of t1

cin>>t1;

cout<<"t2 (s): ";

//Input value of t2

cin>>t2;

//Formula for d1

float d1 = (u\*t1)+(0.5\*a\*t1\*t1);

//Formula for d2

float d2 = (u\*t2)+(0.5\*a\*t2\*t2);

//Formula for d

float d = d2-d1;

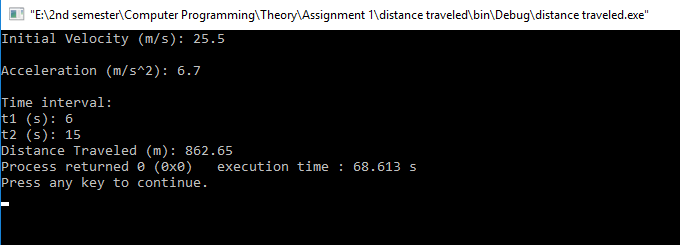
//Display distance covered in given time interval

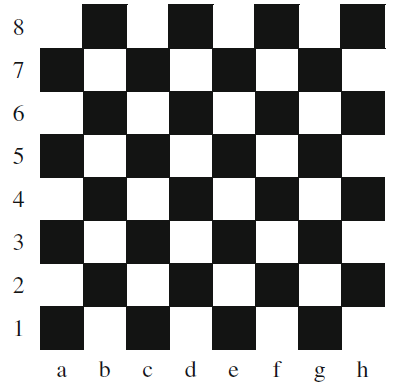
cout<<"Distance Traveled (m): "<<d;

return 0;

}

### **Output (Compilation, Debugging & Testing):**



1. Positions on a chess board are identified by a letter and a number. The letter identifies the column, while the number identifies the row, as shown below:

Write a program that reads a position from the user. Use an if statement to determine if the column begins with a black square or a white square. Then use modular arithmetic to report the color of the square in that row. For example, if the user enters a and 1 then your program should report that the square is black. If the user enters d and 5 then your program should report that the square is white. If a user enters a wrong position display an error message stating “Invalid Position”.

**Code:**

#include <iostream> /\* library for

writing the output to console window\*/

using namespace std;

int main()

{

//Variable declaration

char col;

int row;

cout << "Enter the column: ";

//Input value of col

cin>>col;

cout << "Enter the row: ";

//Input value of row

cin>>row;

//when column entered is a,c,e or g

if(row<=8 && row>=1 && (col=='a' || col=='c' || col=='e' || col=='g'))

{

//Condition for even rows

if(row%2==0)

cout<<"This square is White";

//Odd rows

else

cout<<"This square is Black";

}

//when column entered is b,d,f or h

else if(row<=8 && row>=1 && (col=='b' || col=='d' || col=='f' || col=='h'))

{

//Condition for even rows

if(row%2==0)

cout<<"This square is Black";

//Odd rows

else

cout<<"This square is White";

}

//Invalid Position

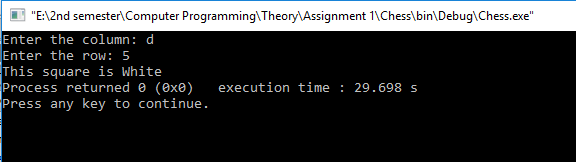
else

cout<<"Invalid Position";

return 0;

}

**Output (Compilation, Debugging & Testing):**



1. A particular cell phone plan includes 50 minutes of air time and 50 text messages for Rs. 15.00 a month. Each additional minute of air time costs Rs. 0.25, while additional text messages cost Rs. 0.15 each. All cell phone bills include an additional charge of Rs. 0.44 to support 911 call centers, and the entire bill (including the 911 charge) is subject to 5 percent sales tax.

Write a program that reads the number of minutes and text messages used in a month from the user. Display the base charge, additional minutes charge (if any), additional text message charge (if any), the 911 fee, tax and total bill amount. Only display the additional minute and text message charges if the user incurred costs in these categories. Ensure that all of the charges are displayed using floating point data type.

**Code:**

#include <iostream> /\* library for

writing the output to console window\*/

using namespace std;

int main()

{

//Variable declaration

int mins,texts;

cout<<"Enter the number of Minutes used this month: ";

//Input value of mins

cin>>mins;

cout << "Enter the number of text messages used this month: ";

//Input value of texts

cin>>texts;

cout<<"\n\n\t\t Cell Phone Monthly Bill:\n\n";

//Display base charge

cout<<"Base Charge: 15.00\n";

//When mins and texts both are greater than 50

if(mins>50 && texts>50)

{

//Formula for additional mins

int additional\_mins = mins-50;

//Formula for additional mins charge

float a\_m\_c = additional\_mins\*0.25;

//Formula for additional texts

int additional\_texts = texts-50;

//Formula for additional texts charge

float a\_t\_c = additional\_texts\*0.15;

//Display additional mins charge

cout<<"Additional Minutes Charge: "<<a\_m\_c<<endl;

//Display additional texts charge

cout<<"Additional Text Message Charge: "<<a\_t\_c<<endl;

//Display 911 Fee

cout<<"911 Fee: 0.44\n";

//Total without tax

float total = 15.00+0.44+a\_m\_c+a\_t\_c;

//Formula for Tax

float tax = (total\*5)/100;

//Display tax

cout<<"Tax: "<<tax<<endl;

//Display Total including tax

cout<<"Total Bill Amount: "<<total+tax<<endl;

}

//When only texts are greater than 50

else if (texts>50)

{

//Formula for additional texts

int additional\_texts = texts-50;

//Formula for additional texts charge

float a\_t\_c = additional\_texts\*0.15;

//Display additional texts charge

cout<<"Additional Text Message Charge: "<<a\_t\_c<<endl;

//Display 911 Fee

cout<<"911 Fee: 0.44\n";

//Total without tax

float total = 15.00+0.44+a\_t\_c;

//Formula for Tax

float tax = (total\*5)/100;

//Display tax

cout<<"Tax: "<<tax<<endl;

//Display Total including tax

cout<<"Total Bill Amount: "<<total+tax<<endl;

}

//When only mins are greater than 50

else if(mins>50)

{

//Formula for additional mins

int additional\_mins = mins-50;

//Formula for additional mins charge

float a\_m\_c = additional\_mins\*0.25;

//Display additional mins charge

cout<<"Additional Minutes Charge: "<<a\_m\_c<<endl;

//Display 911 Fee

cout<<"911 Fee: 0.44\n";

//Total without tax

float total = 15.00+0.44+a\_m\_c;

//Formula for tax

float tax = (total\*5)/100;

//Display tax

cout<<"Tax: "<<tax<<endl;

//Display total bill including tax

cout<<"Total Bill Amount: "<<total+tax<<endl;

}

//When mins and texts are less than or equal to 50

else

{

//Display 911 Fee

cout<<"911 Fee: 0.44\n";

//Total without tax

float total = 15.00+0.44;

//Formula for tax

float tax = (total\*5)/100;

//Display tax

cout<<"Tax: "<<tax<<endl;

//Display total including tax

cout<<"Total Bill Amount: "<<total+tax<<endl;

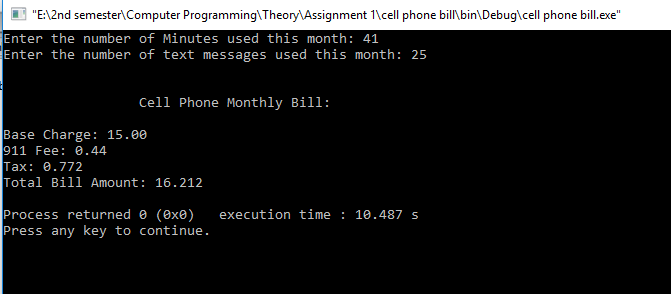
}

return 0;

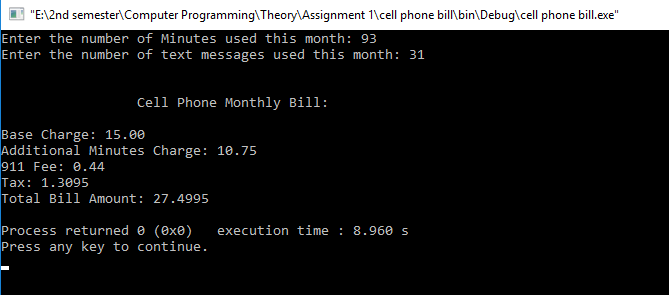
}

**Output (Compilation, Debugging & Testing):**

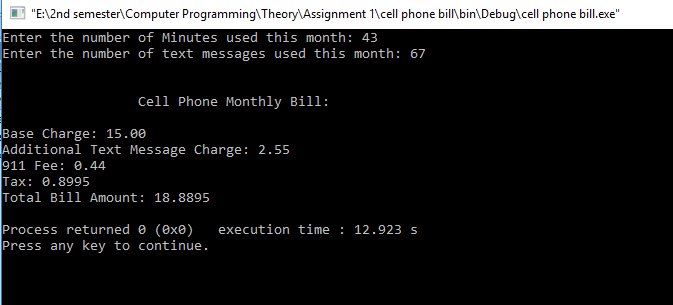
**No additional Texts and Minutes:**

****

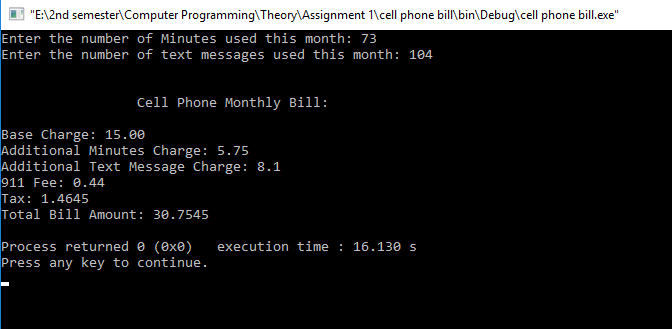
**Additional Minutes Only:**

****

**Additional Texts Only:**

****

**Additional Minutes and Texts:**

****

1. Print the value of y for given x=2 & z=4 and analyze the output.

(a). y = x++ + ++x;

**Code:**

#include <iostream>

using namespace std;

int main()

{

//Variable declaration

int x=2,z=4;

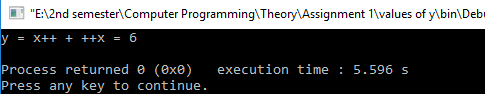
int y= x++ + ++x;

cout <<"y = x++ + ++x = "<< y << endl;

return 0;

}

**Output (Compilation, Debugging & Testing):**



**Analysis:**

Both Pre-increment and Post-increment operators are used here, Pre-increment (++x) operator increments the variable before evaluating it and Post-increment (x++) operator evaluates the variable before incrementing it. In the given program x is initialized to 2. X++ returns 2 and then increments it to 3 and ++x increments x to 4 and returns 4. So y= x++ + ++x = 2 + 4= 6.

(b). y= ++x + ++x;

**Code:**

#include <iostream>

using namespace std;

int main()

{

//Variable declaration

int x=2,z=4;

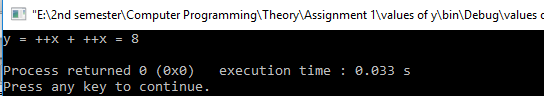
int y= ++x + ++x;

cout <<"y = ++x + ++x = "<< y << endl;

return 0;

}

**Output (Compilation, Debugging & Testing):**

****

**Analysis:**

This Program shows unexpected behavior, the expected output was 7 but the output is 8. The reason for this unexpected behavior is that there is no order of precedence in increment operators.

++ of the first ++x increments x to 3 without returning its value and ++ of the second ++x increments x to 4 after that both the ++x operators return 4. So, y=++x + ++x=4+4=8.

(c). y= ++x + ++x + ++x;

**Code:**

#include <iostream>

using namespace std;

int main()

{

//Variable declaration

int x=2,z=4;

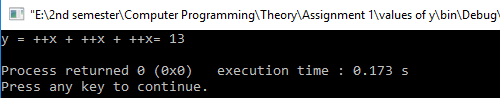
int y= ++x + ++x + ++x;

cout <<"y = ++x + ++x + ++x= "<< y << endl;

return 0;

}

**Output (Compilation, Debugging & Testing):**

****

**Analysis:**

As discussed in the previous problem, the first two ++x operators both return 4 and add them together, the third ++x increments x to 5 and returns its value. So, y=++x + ++x + ++x = 4+4+5=13.

(d). y = x>z;

**Code:**

#include <iostream>

using namespace std;

int main()

{

//Variable declaration

int x=2,z=4;

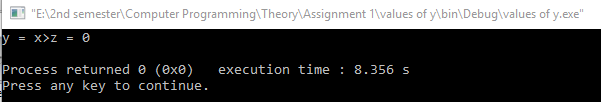
int y= x>z;

cout <<"y = x>z = "<< y << endl;

return 0;

}

**Output (Compilation, Debugging & Testing):**

****

**Analysis:**

In this program we have initialized x to 2 and z to 4, as 2 is less than 4 so the statement x>z is false and returns 0.

(e). y= x>z? x:z;

**Code:**

#include <iostream>

using namespace std;

int main()

{

//Variable declaration

int x=2,z=4;

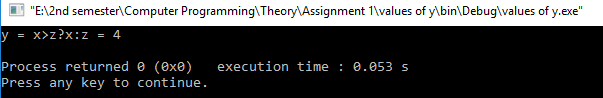
int y= x>z?x:z;

cout <<"y = x>z?x:z = "<< y << endl;

return 0;

}

**Output (Compilation, Debugging & Testing):**

**Analysis:**

The ternary operator ( ? :) executes the statement before the colon if the condition is true and it executes the statement after the colon if the condition is false. As x>z is false so value of z is assigned to y. Hence the output is 4.

(f). y = x&z;

**Code:**

#include <iostream>

using namespace std;

int main()

{

//Variable declaration

int x=2,z=4;

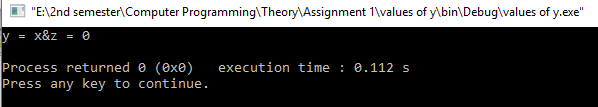
int y= x&z;

cout <<"y = x&z = "<< y << endl;

return 0;

}

**Output (Compilation, Debugging & Testing):**



**Analysis:**

The bitwise AND operator (&) takes the AND of binary x and z.

X= (2)10 = (0010)2 and Z= (4)10 = (0100)2

Y= 0010

&0100

0000

So, 0 is assigned to y.