Air University (Multan Campus)

Dept. Computer Sciences, BSCS-V Fall'2017

Assignment# 3

BugyCode and ProgrammingMistakes.

Subject: Software Engineering

Submitted to:

Sir Ahmad Mohsin

Submitted by:

Umamah Ayyaz Ansari (153174)

Date of Submission:

1st October, 2017

BugyCode:

Issue:

```
// Read code below and try to identify any bug and then fix it.

int main()
{
	cout << "This program will crash"
		<< endl;

	int nSum;
	int nNums;

	// accumulate input numbers until the
	// user enteres a negative number, then
	// return the average
	nNums = 0;
	while(true)
	{
		// enter another number to add
		int nValue;
		cout << "Enter another number:";
		cin >> nValue;
		cout << endl;

		// if the input number is negative...
		if(nValue < 0)
		{
			// ... then output the average
			cout << "Average is: "
				<< nSum/nNums
				<< endl;
			break;
		}

		// not negative, add the value to
		// the accumulator
		nSum += nValue;
	}

	cin.ignore(10000, '\n');
	return 0;
}
```

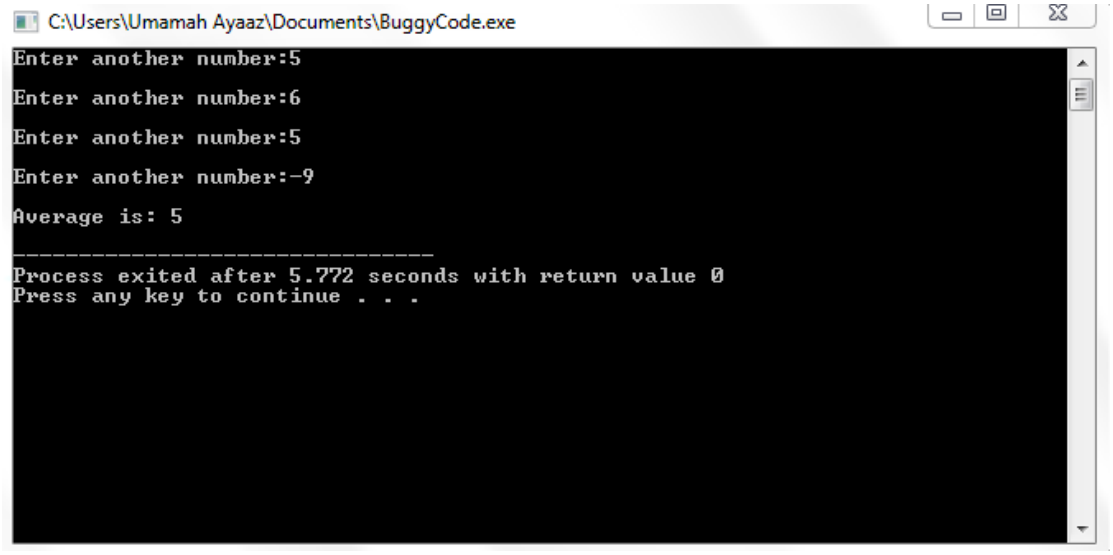
Correction:

```
// Correction: header are not included
// So Including header and “std” library:

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

```
int main()
{
	// cout << "This program will crash"
	//	<< endl;

	int nSum;

	// Correction: initializing the nSum with ZERO:
	nSum=0;

	int nNums;

	// accumulate input numbers until the
	// user enters a negative number, then
	// return the average
	nNums = 0;
	while(true)
	{
		// enter another number to add
		int nValue;
		cout << "Enter another number:";
		cin >> nValue;
		cout << endl;

    while (cin.fail())   // Input Validation: Accept only Integers.
		{
			cin.clear();
			cin.ignore();
			cout << "Invalid Input. Enter an Integer as a Number: " << endl;
			cin >> nValue;
		}

		// if the input number is negative...
		if(nValue < 0)
		{
			// ... then output the average
			cout << "Average is: "
				<< nSum/nNums
				<< endl;
			break;
		}

		// not negative, add the value to
		// the accumulator

		nSum += nValue;

//*****************************************************************
// The problem is nNums is remained ZERO throughout the execution. According to the
// formula of "Average", nNums is a Denominator.
```

```
// Why program is keep on crashing?
// Answer: because nNums is zero throughout the execution of program. Anything divided
// by ZERO is infinity thats why the program is crashing.
// Correction: nNums should be added by 1 everytime a positive number is entered in the
// nValue.

// Why we have to added 1 to nNums everytime a positive number is entered in the nValue?
// because of the formula of "AVERAGE", nSum should be divided by no. of numbers entered
// i.e. nNums
//*********************************************************************

                    nNums+= 1; // can also be written as nNum++


}
        cin.ignore(10000, '\n');
        return 0;
}
```

Output:



ProgrammingMistakes:

1. Undeclared Variables:

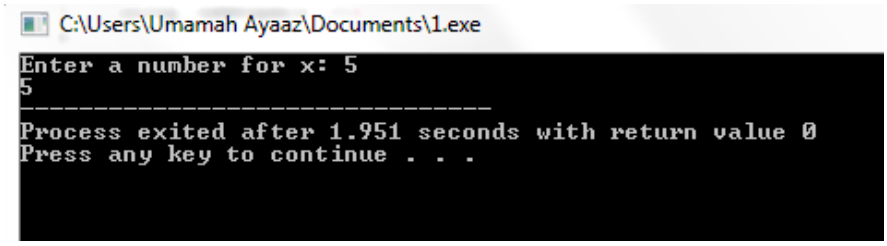
Issue:

```
intmain()
        {
        cin>>x;
        cout<<x;
        }

        Question: "Huh? Why do I get an error?"
```

Correction:

```
//*********************************************************************
// Question: "Huh? Why do I get an error?"
// Answer: because of un-declared Datatype of x
//*********************************************************************

#include <iostream>
using namespace std;

int main()
{
        // Problem: "x" is not declared in this scope
        // Correction: Let's declare x by defining the data-type:
        int x;

        //Guiding the user to enter a value:

        cout <<"Enter a number for x: ";
cin>>x;
cout<<x;
return 0;
}
```

Output:



2. Uninitialized variables:

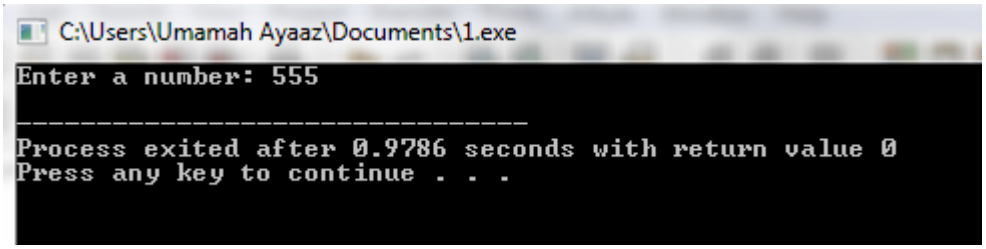
Issue:

```
intcount;
            while(count<100)
            {
            cout<<count;
            }
            "Why doesn't my program enter the while loop?"
```

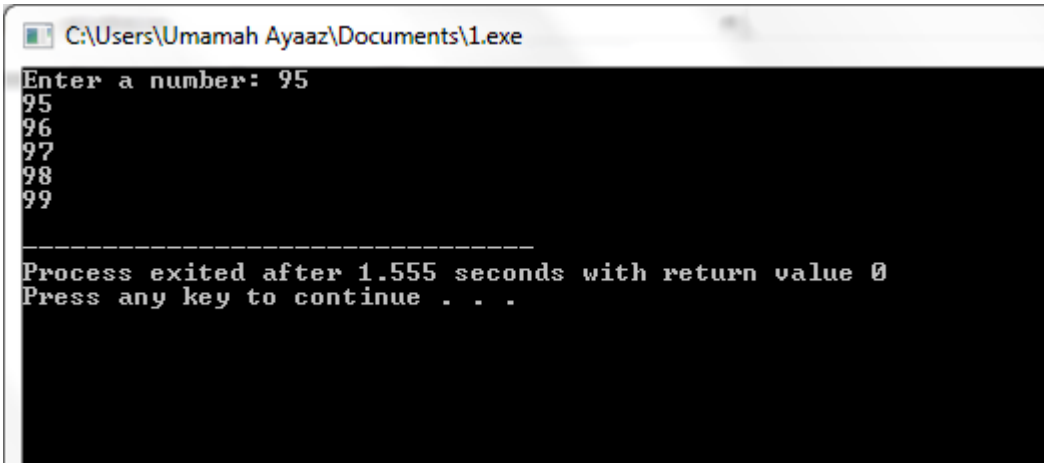
Correction:

```
//*********************************************************************
// Question: "Why doesn't my program enter the while loop?"
// Answer: Because while loop contains "count" and count's value is unknown.
//*********************************************************************


#include <iostream>
using namespace std;

int main ()
{
        int count;

        // The count is declared as "int" but the value of count is unknown
        // that's why "while" loop does not know the value of "count" and gives an error

        // Solution: Let's take the value of "count" from the user

        cout << "Enter a number: ";
        cin >> count;

  while(count<100)
                    // If the "count" is less than 100, the program will enter the "while" loop
                    // otherwise the program will be terminated.
{
  cout<<count;

  // Problem: While loop is an endless loop here.
  // Solution: To end the while loop, "count" should be greater than 100.
// to end the loop, let's keep on adding "1" in count.

        cout << endl;
        count+=1;
}
 return 0;
}
```

Output:



OR



---

3. Setting a variable to an uninitialized value :

Issue:

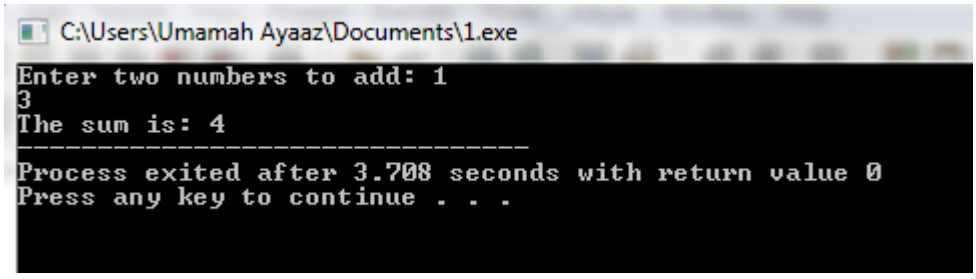
```
inta,b;
        int sum=a+b;
        cout<<"Enter two numbers to add: ";
        cin>>b;
        cout<<"The sum is: "<<sum;
        When Run:
        Enter two numbers to add: 1 3
        The sum is: -1393
        "What's wrong with my program?"
```

Correction:

```
//*********************************************************************
//  Question: "What's wrong with my program?"
// Answer:
// -- First problem was the definition of sum, which was solved by
// simple datatype declaration.
// -- Second problem was the formula of "sum". It should not be defined in the start.
// -- Third problem was the program was asking for the value of "b" only from the
user,
```

```
// which is solved by asking the value of both "a" and "b".
// -- Fourth problem was the formula for "sum", which is solved by defining it before
// displaying the sum.
//*********************************************************************


#include <iostream>
using namespace std;

int main()
{
        int a, b;

// Lets define "sum" as "int sum" instead of "int sum=a+b" :
 int sum;

cout<<"Enter two numbers to add: ";

// Problem: Sum= a + b. where the program is asking from the users to enter a value
for "b" only
// Correction: Here "a" should also be entered before "b"
cin >> a;
cin >> b;

// Lets write the formula for sum here i.e. before the displaying of sum:
    sum = a+b;

// Now the sum will be displayed:
cout<<"The sum is: "<< sum;

return 0;
}
```

Output:



4. Using a single equal sign to check equality:

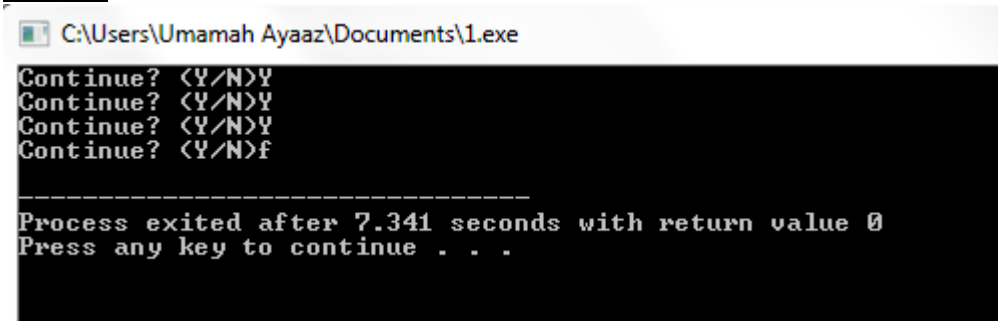
Issue:

```
charx='Y';
                while(x='Y')
                {
                //...
                cout<<"Continue? (Y/N)";
                cin>>x;
                }


                "Why doesn't my loop ever end?"
```

Correction:

```
//*************************************************************************
//Question: "Why doesn't my loop ever end?"
// Answer: Because the selection of the operation was wrong.
// Solution is usage of Equal-to operator "==" instead of
// Assignment operator "="
//*************************************************************************

#include <iostream>
using namespace std;

int main()
{
        char x= 'Y';

        // Problem: Wrong operator is used here.
        // Solution: using "==" operator instead of Assignment operator i.e. = :
while(x=='Y')
{
//...
cout<<"Continue? (Y/N)";
cin>>x;
}
return 0;
}
```

Output:



5. Undeclared Functions:

Issue:

```
intmain()
        {
        menu();
        }
        void menu()
        {
        //...
        }
        "Why do I get an error about menu being unknown?"
```

Correction:

```
//*********************************************************************************
// Why do I get an error about menu being unknown?
// Answer: Because compiler goes through the program exactly once.
// Functions would have to be declared before they're used; prototypes let us
//declare a function without implementing its body until later.
// Similarly we introduced the prototype of the function “menu”.
// OR the another solution is to define whole function before “main()” instead of
// defining at the end of the Program i.e. after “main()”.
//*********************************************************************************

#include <iostream>
using namespace std;

// Problem: Compile says the function "menu" was not declared.
 // Solution: lets introduce a Prototype of the function "menu" before calling
  // it in "main()"

void menu();  // Declared the function here.

int main()
{
menu();
Return 0;
}

void menu()
{
//...
}
```

OR

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

void menu()
```

```
{
//...
}

int main()
{
menu();
Return 0;
}
```

---

6. Extra Semicolons:

Issue:

```
  for(intx=0;x<100;x++);
                                cout<<x;
```

Question: "Why does it output 100?"

Correction:

```
#include <iostream>

using namespace std;

int main()
{
    // A semicolon at the end of a “for” loop statement is against the syntax.
    // Solution: Remove the semicolon at the end of “for” loop statement.

     for(int x=0; x<100; x++)
     cout<<x;

return 0;

}
```

---

7. Overstepping array boundaries:

Issue:

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

int main()
{
int array[10];
//...
For (int x=1; x<=10; x++)
cout<<array[x];

return 0;
}
Why doesn't it output the correct values?"
```

Correction:

```
//*********************************************************************************************
// Question: Why doesn't it output the correct values?"
// Answer: Because index of an Array always start from "0" not from "1".
// Solution: Initializing "x" with “0 “instead of " 1 ".
//*********************************************************************************************

#include <iostream>
using namespace std;

int main()
{
int array[10];
//...
For (int x=0; x<=10; x++)   // Here x is initialized with 0 instead of 1.
cout<<array[x];

return 0;

}
```

---

8. Integer division:

Issue:

```
double half = 1/2;
```

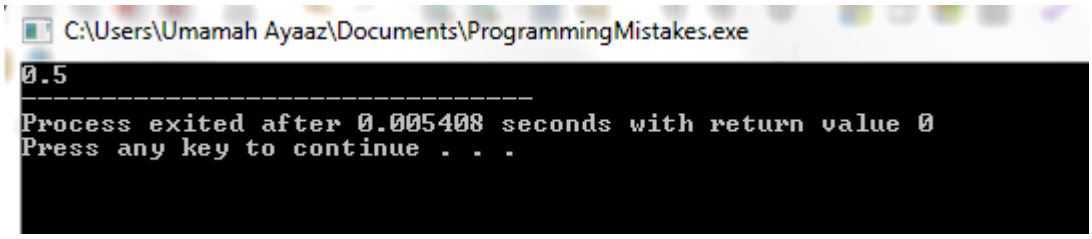
This code sets half to 0 not 0.5! Why? Because 1 and 2 are integer constants.

Correction:

```
//*********************************************************************************************
// Question: This code sets half to 0 not 0.5! Why? Because 1 and 2 are integer
```

```
// constants.
// Answer: The numbers “1” and “2” are integer and half is defined as a double.
// By giving half simple 1 and 2 we are limiting their answers. It’s better to use “1.0”
// and “2.0” instead of “1” and “2” respectively.
//*********************************************************************************************


Double half = 1.0/2.0;
```

Output:



---

9. Variable Name Styles:

Issue:

Take a look at the below program. Can you see anything wrong?

```
#include <stdio.h>
#include <string.h>

main()
{
char CatName[20] = "fluffy";
char dogName[20] = "fido";
char rat_Name[20] = "fester";
int Catage = 3;
int dogs_age = 4;
int ratage = 1;
char myPet[20];
int itsAge;

strcpy(myPet,rat_Name);
itsAge = ratage;
printf("My pet is %s\n",myPet);
}
```

Question: Besides the obviously bizarre choice of having a pet rat, do you notice anything?

Answer (Correction):

```
//*********************************************************************************************
// Question: Besides the obviously bizarre choice of having a pet rat, do you notice
```

```
// anything?
// Answer: The code is fine (correct). rat_Name was copied into myPet and itsAge is
// also defined as ratage.
//*********************************************************************************
```

---

10. Misusing the && and || operators:

Issue:

```
int value;
do
{
//...
value=10;
}while(!(value==10) || !(value==20))
```

Question: "Huh? Even though value is 10 the program loops. Why?"

Correction:

```
//*********************************************************************************
// Question: "Huh? Even though value is 10 the program loops. Why?"
// Answer: First issue is the “while” condition is used without following the proper
// syntax. In do-while conditional loop, a semicolon is essential after “while”
// Now to resolve the main issue which was asked, we have to know the usage of
// both “||” and “&&” operator.
// According to the given scenario, we should use “||” instead of “&&”.
//*********************************************************************************

int value;
do
{
//...
value=10;
}while(!(value==10) && !(value==20));
```
