C++ Indentation and Error Detection

Kailash JP-2017103018

Surenthernath J-2017103601

Github Link:

https://github.com/kailashjp/Code_Indentor_Error_Detector

Objective :

Given a C++ program as input, the C++ code Indentation and Error detector displays intended C++ program with detected errors.

Functionalities available:

- Declaring and initializing variables
- Operations : Arithmetic, logical, assignment ,relational
- Reading input and printing output
- Statement completion
- Functions and procedures

Tools used:

- Lex
- Yacc

Implementation:

This compiler has been constructed using lex and yacc . The keywords/ tokens are declared in the lex file (intendation_error.l). The productions and rules for displaying intended c++ code are implemented in yacc file (intendation_error.y).

Commands:

```
lex indentation_error.l
yacc -d indentation_error.y
gcc lex.yy.c y.tab.c
./a.out test1
```

Sample Input-1:

```
#include<iostream.h>
int main()
{return ;}
int sample(float f)
{
int x=10;
int y=20;
int z;
int i ;
if(x<y)
{z=5;
}
for(i=0;i<10;i++){printf("Hie");}
return 0;
}
```

Sample Output -1:

```
#include<iostream.h>
int main()
{
	return;
}
int sample(float f)
{
	int x=15;
		int y=30;
			int z;
		if(x>y || x==5)
		{
			z=5;
		}
		for(i=0;i<10;i++)
		{
			printf(“Hie”);
		}
		return 0;
}
```

Sample Input-2:

```
#include<iostream>
#include<string.h>
int main(){c=0;
return ;}
int sample2(float x,float y){h=0;
for(i=0;i<1;i++){h++;}
return h;
}
```

Sample Output-2:

```
#include<iostream>
#include<string.h>
int main()
{
                    c=0;
                    return;
}
int sample2(float x,float y)
{
                        h=0;
                            for(i=0;i<1;i++)
```

```
                {
                    h++;
                }
            return h;
}
```