Report on Programming Asssignment # 0
Toy Cool Programs

Name : Shivashish Suman
Roll No. : CS17BTECH11037

Correct Codes:

1.In the MIPS code of first correct code, we see that print statements, function return types are present in the str_const which are grouped together in class_namTab. Class Main stores its member functions name in Main_dispTab. Each function is first allocated space in stack using ‘addiu’ .Then there are multiple store and load store instructions. We have used ‘<=’ in if-statement in fibonacci function which is translated to ‘ble’ (branch if less than or equal to) in MIPS. We also used condition ‘x=0’ in if-statement which is translated to ‘beqz’ (branch if equal to zero). It also contains the instruction ‘jal’ which is jump and link in MIPS.It also contains ‘bne’ instruction which is branch if not equal.The MIPS code also contains ‘jalr’ which is same as ‘jal’ except the address of the subroutine now comes from a register.

2.In the second MIPS code of program of ‘sumofcubes’, we see that that there is ‘mul’ instruction used to multiply two values as used in sumofcubes function and there are multiple load and store operations implemented using ‘lw’ and ‘sw’ instruction.There is also ‘b’ instruction used which specifies unconditional branching. ‘mov’ instruction is used to move contents of one register to another. MIPS code contains‘li’ instruction which loads 32-bit immediate into a register.

3. In the third program which checks whether a number is divisible by 5 or not, the MIPS code has almost the same instructions as in the first two programs.We also observe that

Bool_init, String_init, Int_init, Main_init, Object_init and IO_init labels are present in all the 5 programs. '.globl sym' declares that symbol sym is global and can be referenced from other files. '.word' stores 32-bit quantities in memory words. In the 'Main.divby5' , we can see that first it uses condition 'x<0' in if-statement by using 'blt' and then there are 'beq' instructions for comparison of the value of 'x'.

4. In the fourth correct program, we have computed the factorial of number. 'Main.factorial' label contains 'ble' instruction which indicates the statement 'x<=0' which is branch if less than or equal to.It checks for 'x=0' using 'beqz' instruction and then goes to given label otherwise it goes unconditionally to other branch using 'b' instruction which indicates else part in if-statement.It has 'sub' instruction which subtracts the value stored in one register from second register.

5.In the fifth correct program, we reverse the given input string. It checks if x is non-zero by using 'bne' instruction and it also has 'beqz' instruction to branch if it is zero.

Incorrect Codes:

1. In the first incorrect code, we try to break the rule for naming the identifiers. We have used '@' in the identifier which is not allowed. We have also started name of object with capital letter which is not allowed.
Cool Compiler shows the following error:
line 5: syntax error at or near TYPEID = Ans
line 23: syntax error at or near ASSIGN
Here 1st error shows that we have made an error while naming the object to be 'Ans' and 2nd error shows that '@' is not allowed in identifiers.

2. In the second incorrect code, we break the rule for strings. Strings are enclosed in double quotes(“ ”). But we have used single quotes(‘ ’).
Compiler shows the following error:
line 9: syntax error at or near ERROR = '
This tells that we have made mistake while using single quotes instead of double quotes.

3.In the third incorrect code, we try to break the rule for comments. Comments are enclosed within (* *) . But we don’t have * in the beginning of comment which gives us the error.
Compiler shows us the following error:
line 3: syntax error at or near '('
This tells us that there should be an * symbol after the opening paranthesis to indicate beginning of multi-line comment.

4.In the fourth incorrect code, we try to break the rule for keyword ‘true’. Rule says that first letter of true and false must be lowercase and the trailing letters may be upper or lower case. But we have written ‘True’ which begins with upper case.
Compiler shows us the following error:
line 7: syntax error at or near TYPEID = True
This tells us that keyword ‘true’ should begin with lower case.

5.In the fifth incorrect code, we try to break the rule for whitespace. In this we don’t put a whitespace between words ‘false’ and ‘fi’. So compiler considers it to be a single word ‘falsefi’.
Compiler shows us the following error:
line 9: syntax error at or near '}'
This shows that compiler does not find any statement for else part and neither it finds the ‘fi’ that marks the end of if-statement.