

Assignment - 1.

The difference between interpreter and compiler :-

COMPILER	INTERPRETER
1. Compiler works on the complete program at once. It takes the entire program/code as input.	Interpreter works line-by-line. It takes one statement at a time as input.
2. Compiler generates intermediate code, called the object code or machine code.	Interpreter does not generate intermediate object code or machine code.
3. Compiler executes conditional control statements (like if-else and switch-case) and logical constructs faster than interpreter.	Interpreter execute conditional control statements at a much slower speed.
4. Compiled programs takes more memory because the entire object code has to reside in memory.	Interpreter does not generate intermediate object code. As a result interpreted programs are more memory efficient.
Teacher's Signature.....	

- | | |
|--|---|
| 5. Compile once and run anytime. Compiled program does not need to be compiled every time. | Interpreted programs are interpreted line-by-line every time they are run. |
| 6. Errors are reported after the entire program is checked for syntactical and other errors. | Errors are reported as soon as the first error is encountered. Rest of the program will not be checked until the existing error is removed. |
| 7. A compiled language is more difficult to debug. | Debugging is easy because interpreter stops and reports errors as it encounters them. |
| 8. Compiler does not allow a program to run until it is completely error free. | Interpreter runs the program from first line and stops execution only if it encounters an error. |
| 9. Compiled languages are more efficient but difficult to debug. | Interpreted languages are less efficient but easier to debug. This makes such languages an ideal choice for new students. |

10. Examples of programming languages that use compilers :- C, C++, COBOL

Examples of programming languages that use interpreters : BASIC, Visual Basic, Python, Ruby, PHP, Perl, Lisp, MATLAB