

# Introduction to computer science and program design assignment



## Case Assignment: Computers, Computer Science, and Computer Scientists

Dry. Walter Makeover Table of Contents Table of Contents 2 Introduction 2 History and Early Programming 3 Early Advances 5 Structured Programming 5 Object-oriented Programming 6 Need for New Paradigms 6 Future Trends 7 Conclusion 7 References 8 Introduction When you think of a computer in today's terms, a number of items can come to mind such as a PDA, laptop or PC, when not long ago computers were huge canines.

Some Cell phones have a computer keypad and can now receive small movie trailers and play video, small games, and even browse the internet. People can even go to web sites and pick which ring tone or display to be sent to their phone or handset's memory. The message is delivered as a binary message over the network. This is an example of people tailoring the device to their specifications and desires. Imagine a society where small kids are thought computer programming and it becomes commonplace in everyday life, where a language is made for the everyday person.

Someday people from all walks of life could design small programs to tailor their PC's and other handheld computer devices to their needs. This day may not be that far away. Computer programming has come a long ways in the past several decades, with the prevalence of the internet and the fast paced information age. Computer programming will likely be streamlined and go through more changes. There have been many emerging technologies used with the Internet. One pattern that is emerging is the use of scripting languages or typesets languages that can be used as an all-purpose language and be platform independent.

This paper will discuss the role of computer programming and where it might lead in the 21st century. It will show some of the early starts of computer programming, major developments, and what types of programs will be more relevant in the 21st century. History and Early Programming Computer programming allows people to give instructions to computers. This is the primary role of computer programming. Charles Babbage, a noted mathematician in England, noticed that many calculations consisted of a series of predictable actions that were constantly repeated.

He thought this process could be automated. He reared something called the difference engine back in 1822 with columns of gears that some recognize as the first modern calculator. L This device used motion as a form of programming. For each calculation gears would have to be changed, and he later worked on the idea of an analytical engine. 2 Augusta Dad Byron worked with Babbage and she was able to see that mathematics would continue to grow and people would be able to use symbols to represent just about anything in the real world. 3 She saw the great potential of the analytical engine or computer.

Another major advance in computer history and programming was with punched arid. Herman Hollering and James Powers developed punched cards and were using them with tabulating machines in 1890. The first major use of them was for the US. Census Bureau. Punched cards allowed for input and memory storage. 4 John Cheer and John Macaulay working with others built a high speed electronic computer in 1942 at the University of Pennsylvania.

This computer used 18000 vacuum tubes and was used to calculate trajectory tables for weapon systems for the military.

<https://assignbuster.com/introduction-to-computer-science-and-program-design-assignment/>

This computer became known as the MANIAC. This computer became recognized as the first successful high speed electronic digital computer and was actively used from 1946 to 1955. 5 At Harvard, Howard Keen and Robert Campbell worked on a high speed digital computer. The idea came to Howard Keen in 1937 while a graduate student at Harvard. It was built by IBM and became known as the Harvard Mark-1. 6 These early computers solved mathematical problems, quite often for the military. The preparation of the operating instructions was done with plugging charts and floorboards.

A couple of major ideas about programming came from John von Neumann. He had the idea of creating small blocks of code and using something he called " Conditional Control Transfer". Instead of using one set of ordered statements, he thought of using subroutines or small blocks of code. Re-use of code has become one of the themes of programming over the years. He also is responsible for the concept of " Shared-program Technique", which means that computer hardware should not have to be re-wired for each new problem, but the hardware should remain simple.

Complex instructions should be used to control the hardware. 7 John von Neumann was a brilliant person in the field of mathematics and held a position at the Institute for Advanced Study in Princeton New Jersey, where he worked with a computer in the 1946 to 1950 timeshare and started writing code for it. It used his stored or shared program concept unlike many of the earlier computers that had to be re-wired for each new problem. This early computer had a total of 29 instructions by July 1952. 8 Early Advances Machine Language is the binary notation that the computers understand.

The programmer had to keep track of where everything was stored in the machine's memory. Later Assembly Language was introduced where mnemonics are used to represent computer operations. Assembly language is where the raw binary code instructions are replaced with symbols such as store, load, and add. The symbols are translated by an assembler to the raw binary machine language. Another major advance in computer programming was the use of a compiler. This allowed for higher level languages.

Languages have become easier to use and closer to spoken language over time.

As a result, we don't have to worry about the machine code. We use a higher level language or one that has some syntax and language in common with the English language. A compiler is used to create the machine code or do the translation to binary. One of the first people who created and used compilers was Grace Hopper. She is considered by many the world's first programmer and developed the first compiler. 9 Structured Programming Structured programming was introduced in the 1950's and one of the best examples of this was Pascal. Structured programming eliminates GOTO statements.

Structured programming is where subunits of code have a single entrance point and a single exit point. There can be sequencing such as do A, then B, then C. Alternation, either do A or do B, and iteration, repeat A until some condition is met. These recommendations prevented spaghetti code. C is an example of a structured programming language. 0 Object-oriented

Programming Procedural and Structured programming focuses on action

words or verbs instead of nouns or objects. However, we live in a world of <https://assignbuster.com/introduction-to-computer-science-and-program-design-assignment/>

objects. Object oriented programming focuses on the objects. Every object has attributes that are defined.

For example, file objects or paycheck objects are some examples.. A file object could have attributes such as size, date, and permissions for example. Object-oriented programming has many languages such as Java, C++, Smalltalk, and Python. Need for New Paradigms While many of advances in computer programming have come to fruition, imputer hardware has outpaced computer programming techniques at an alarming rate. This is a basic problem with computer programming. It is clear that some new methods or paradigms will have to be established if that problem will be solved.

In a book written by Martin K Gay, called " Recent Advances & Issues in Computers", the author points out that some people, including Dry. Lynn Stein of MIT promotes different and new ways of looking at programming. Traditional ways are not ideal anymore. It is essential that programming elements becomes easier and more user- friendly. 11 . An example off program language that is more user-friendly, but yet till powerful enough for expert users is Python. This language was designed for both novices and experts. Future Trends What are the trends in programming for the future?

Robert Martin gave an interview to Toward in 2001, and predicted that we will see an end to statically typed languages or type-safe languages such as Pascal, C++, Java, Eiffel, and DAD. In these languages, the variables must be declared before you can use them. He thinks that we will see more dynamically typed languages such as Smalltalk, Ruby, and Python. His

argument is that the compile time can become very long with type-safe languages. With C++, after a million lines of code, compile time starts to rise drastically. Whereas dynamically typed languages can compile instantaneously.

## 2 Conclusion

When you look at children of today one can only wonder what is on the horizon for advances in programming. With the explosion of the internet age and where people from all walks of life are now using computers, it becomes certain that new ways of programming will come to existence. We have seen how computer programming started and how it was mostly confined to the Universities and mathematicians of the 1950s through the 1960s. We have seen how it evolved from machine, to assembly, to higher language forms. From structured programming to object-oriented programming.