



CP

A. Y.
Surywanshi

Outline

Course
Information

UNIT
1: Process of
programming:

BTES104 Computer Programming in C

Avinash Y. Surywanshi

KCES's College of Engineering and Management,
Jalgaon - 425001, Maharashtra, India

February 9, 2021



BTES104 Computer Programming in C

CP

A. Y.
Surywanshi

Outline

Course
Information

UNIT
1:Process of
programming:

1 Course Information

2 UNIT 1:Process of programming:



CP

A. Y.
Surywanshi

Outline

Course
Information

UNIT
1:Process of
programming:

1 Course Information

2 UNIT 1:Process of programming:



Course Objectives

CP

A. Y.
Surywanshi

Outline

Course
Information

UNIT
1: Process of
programming:

- 1 To give a broad perspective about the uses of computers in engineering industry and C Programming.
- 2 To develop the basic concept of algorithm, algorithmic thinking and flowchart.
- 3 To apply the use of C programming language to implement various algorithms and develops the basic concepts and terminology of programming in general.
- 4 To make familiar the more advanced features of the C language.
- 5 To identify tasks in which the numerical techniques learned are applicable and apply them to write programs and hence use computers effectively to solve the task.



Course Outcomes

CP

A. Y.
Surywanshi

Outline

Course
Information

UNIT
1: Process of
programming:

- 1 Gain a broad perspective about the uses of computers in engineering industry and C Programming.
- 2 Develop the basic concept of algorithm, algorithmic thinking and flowchart.
- 3 Apply the use of C programming language to implement various algorithms and develops the basic concepts and terminology of programming in general.
- 4 Use the more advanced features of the C language.
- 5 Identify tasks in which the numerical techniques learned are applicable and apply them to write programs and hence use computers effectively to solve the task.



Syllabus

CP

A. Y.
Surywanshi

Outline

Course
Information

UNIT
1: Process of
programming:

Unit 1: Process of programming:

(4 Lectures)

Editing, Compiling, Error Checking, executing, testing and debugging of programs. IDE commands. Eclipse for C Program development, Flowcharts, Algorithms. (4 Lectures)

Unit 2: Types, Operators and Expressions:

(4 Lectures)

Variable names, Data types, sizes, constants, declarations, arithmetic operators, relational and logical operators, type conversions, increment and decrement operators, bitwise operators, assignment operators and expressions, conditional expressions precedence and order of evaluation.



Syllabus

CP

A. Y.
Surywanshi

Outline

Course
Information

UNIT
1: Process of
programming:

Unit 3: Control Flow:

(4 Lectures)

Statements and Blocks. If-else, else-if switch Loops while and for, do-while break and continue goto and Labels. Functions and Program Structure: Basic of functions, functions returning non-integers external variables scope rules.

Unit 4: Arrays in C:

(4 Lectures)

Initializing arrays, Initializing character arrays, multidimensional arrays.

Unit 5: Structures C:

(4 Lectures)

Basics of structures, structures and functions arrays of structures,
Pointer in C. Pointers to integers, characters, floats, arrays, structures.

Special Note: Topic of Pointers in C is only for lab exercises and not for end semester examinations.



Teaching and Evaluation Scheme

CP

A. Y.
Surywanshi

Outline

Course
Information

UNIT
1: Process of
programming:

Semester I									
Course Code	Course Title	Teaching Scheme			Evaluation Scheme				
		L	T	P	CA	MSE	ESE	Total	Credit
BTES104	Computer Programming in C	3	-	-	20	20	60	100	2
	Computer Programming in C Lab	-	-	2	60	-	40	100	1



Reference/Text Books

CP

A. Y.
Surywanshi

Outline

Course
Information

UNIT
1: Process of
programming:

1. Brain W. Kernighan & Dennis Ritchie, The C Programming Language, Prentice Hall, 2nd Edition, 1988.
2. R. S. Bichkar, Programming with C, Orient Blackswan, 1st Edition, 2012.
3. Herbert Schildt, C the Complete Reference, McGraw-Hill Publication, 2000.
4. Balguruswamy, Programming in C, PHI.
5. Yashwant Kanitkar, Let Us C, PHI



CP

A. Y.
Surywanshi

Outline

Course
Information

UNIT
1:Process of
programming:

1 Course Information

2 UNIT 1:Process of programming:



What is C

CP

A. Y.
Surywanshi

Outline

Course
Information

UNIT
1: Process of
programming:

- C is general purpose programming language.
- It was developed by AT & T Bell lab of USA.
- It was designed and written by **Dennis M. Ritchie**.
- c has only **32 keywords**.
- Its straight lies in its built in function.
- C is Middle level language, some authors refers C as High level language.
- It is a portable language i.e machine independent.



Facts about C

CP

A. Y.
Surywanshi

Outline

Course
Information

UNIT
1: Process of
programming:

- C was invented to write an operating system called UNIX.
- C is a successor of B language which was introduced around the early 1970s.
- The language was formalized in 1988 by the American National Standard Institute(ANSI) .
- UNIX OS was completely written in C.
- Today C is a most widely used and popular system programming language.
- Most of the software have been implemented using C.
- Today's most popular Linux OS and RDBMS MySQL have been written in C.



Editing, Compiling, Error Checking, executing, testing and Debugging

CP

A. Y.
Surywanshi

Outline

Course
Information

UNIT
1: Process of
programming:

- To type your C program you need program called **Editor**.
- We generally write a computer program using a high-level language.
- computer does **not understand** high-level language.
- program written in 0 and 1 in binary ,called the **machine code**.
- A program written in high-level language is called a **source code**.
- We need to convert the source code into machine code and this is accomplished by compilers and interpreters.
- Programming Languages uses following in built programs.
 - Compiler
 - Interpreter
 - Linker



Difference between Interpreter and Compiler

CP

A. Y.
Surywanshi

Outline

Course
Information

UNIT
1: Process of
programming:

Interpreter	Compiler
Translates program one statement at a time.	Scans the entire program and translates it as a whole into machine code.
It takes less amount of time to analyze the source code but the overall execution time is slower.	It takes large amount of time to analyze the source code but the overall execution time is comparatively faster.
No intermediate object code is generated, hence are memory efficient.	Generates intermediate object code which further requires linking, hence requires more memory.
Continues translating the program until the first error is met, in which case it stops. Hence debugging is easy.	It generates the error message only after scanning the whole program. Hence debugging is comparatively hard.
Programming language like Python, Ruby use interpreters.	Programming language like C, C++ use compilers.



Types of Error

CP

A. Y.
Surywanshi

Outline

Course
Information

UNIT
1: Process of
programming:

- Syntax Error or Compiler Time Error
- Run Time Error or Logical Errors



Program Execution

CP

A. Y.
Surywanshi

Outline

Course
Information

UNIT
1: Process of
programming:

Assuming that you are using a Turbo C or Turbo C++ compiler here are the steps that you need to follow to compile and execute your first C program

- Start the compiler at C:\prompt.
- The compiler (TC.EXE is usually present in C:)
- Select New from the File menu.
- Type the program.
- Save the program using F2 under a proper name (say Program1.c).
- Use to Alt + F9 compile and Ctrl + F9 execute the program.
- Use Alt + F5 to view the output.



Program Testing

CP

A. Y.
Surywanshi

Outline

Course
Information

UNIT
1: Process of
programming:

- Testing is the process of finding errors ("bugs") in a program.
- Testing can increase your confidence that a program is error-free.
- Testing can find the presence of errors, but, in general, cannot prove the absence of errors.
- Testing small programs is much easier than testing large programs.
- When testing code with branches, provide inputs that test all code sections of the if statement.



Program Debugging

CP

A. Y.
Surywanshi

Outline

Course
Information

UNIT

1: Process of
programming:

- Debugging is the process of locating and correcting errors in a program.
- Debugging is problem-solving and often can be very challenging.
- Thinking carefully about your program is often the best first step when debugging.



Program Debugging

CP

A. Y.
Surywanshi

Outline

Course
Information

UNIT
1: Process of
programming:

You can help minimize your time spent debugging by:

- Starting with a good program design,
- Coding carefully, and
- Testing your program as you write.
- The development of good debugging skills comes from experience. Therefore, make sure you learn from your mistakes.



Basic Structure of C Program

CP

A. Y.
Surywanshi

Outline

Course
Information

UNIT

1: Process of
programming:

```
/* Program to illustrate general structure of C program */
/*Global variables declaration; */
#include<stdio.h>
#include<conio.h>
Function prototype declaration;
void main()
{
    /* body of program – it includes declaration of variables, constants etc.
    statements, expressions, loops, function declaration */
    getch();
}
Function definition (parameter list) /* Function is Optional */
{
    /* Body of Function */
}
```

Figure: Basic Structure of C Program



First C Program

CP

A. Y.
Surywanshi

Outline

Course
Information

UNIT
1: Process of
programming:

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

```
int main()
```

```
{
```

```
    printf("Hello world!\n");
```

```
    return 0;
```

```
}
```



Program for addition of two numbers

CP

A. Y.
Surywanshi

Outline

Course
Information

UNIT

1: Process of
programming:

```
/* Program for addition of two numbers*/
#include<stdio.h>
#include<conio.h>
void main()
{
    int a = 10, b = 20, c;
    c=a+b;
    clrscr();
    printf("Addition of a \& b is = \%d",c);
    getch();
}
```



Things Included in program

CP

A. Y.
Surywanshi

Outline

Course
Information

UNIT
1: Process of
programming:

- Comments
- Preprocessor Directive
- Header files
- main () Function
- Braces pair
- Declaration of variables and constants
- Expression
- clrscr ();
- getch(); get character



Examples of header files

CP

A. Y.
Surywanshi

Outline

Course
Information

UNIT
1: Process of
programming:

```
#include <string.h> - For String operations  
#include <math.h> - For Mathematical operations  
#include <conio.h> - For console input output  
operations like getch(), clrscr()
```




Format Specifier

CP

A. Y.
Surywanshi

Outline

Course
Information

UNIT
1: Process of
programming:

Format Specifier	Value Type
%d	Integer value
%f	Float value
%c	Character value
%s	String value
%ld	Long integer value
%lf	Double value
%Lf	Long double values
%u	Unsigned value



CP

A. Y.
Surywanshi

Outline

Course
Information

UNIT
1: Process of
programming:

Thank You...

Email : avin_surya@yahoo.com