EXERCISE

Q1. Select the best answer for the following MCQs.

i. What defines the rules of valid statements in programming?

A. Compiler
B. Interpreter
C. Syntax
D. Semantic

ii. Which language is directly understood by the computer?

A. Machine language
B. Assembly language
C. High-level language
D. C language

iii. When was C language developed?

A. Late 1960s
B. Early 1970s
C. 1980s
D. 1990s

iv. Who developed Java language?

A. Dennis Ritchie
B. Microsoft
C. Sun Microsystems
D. IBM

v. **What is the other word used for Reserved Words?**

A. Compiler words

B. Keywords

C. Special programming words

D. Mnemonics

vi. **How many bytes are set aside by the compiler for a variable of type int?**

A. 2

B. 3

C. 4

D. 5

vii. **How many bytes are set aside by the compiler for a variable of type float?**

A. 2

B. 3

C. 4

D. 5

viii. **What is the range of numbers that can be stored in a variable of type double float?**

A. -32,768 ~ +32,767

B. $10^{-38} \sim 10^{38}$

C. $10^{-308} \sim 10^{308}$

D. $10^{-4932} \sim 10^{4932}$

ix. **Which program translates high-level language into machine language?**

A. Compiler
B. Linker
C. Loader
D. Debugger

x. **Which software helps in finding and removing errors in programs?**

A. Linker
B. Text Editor
C. Loader
D. Debugger

Answers

i. C	**ii. A**	**iii. B**	**iv. C**	**v. B**
vi. A	**vii. C**	**viii. C**	**ix. A**	**x. D**

SHORT QUESTIONS

Q2. Give short answers to the following questions:

i. Define computer program.

Ans: Computer Program:

A computer program is a set of instructions (statements) written in a programming language to solve a particular problem and achieving specific results. Any task performed by a computer is controlled by a set of instructions that are executed by the microprocessor. A large variety of programming languages have been developed for writing computer

programs to use the computer as a problem-solving tool. Each statement of a programming language has syntax and semantic.

ii. Differentiate between syntax and semantic.

Ans: Syntax:

Syntax refers to the rules of a programming language according to which statements of a program are to be written. It describes the way to write correct statements in a program. Syntax of a programming language is similar to the grammar of a natural language.

For example, an assignment statement consists of a variable and an expression separated by equal sign as an assignment operator. This is the syntax of assignment statement and it can be expressed as given below

Variable=expression

Semantic:

Semantic gives meaning to statements of a programming language. It describes the sequence of operations to be performed by a computer when executing the statements of a computer program.

For example, in the assignment statement

Sum = a + b ;

The semantic of the statement is to perform the expression, that is, add the values stored in variables a and b and then store the result in variable sum.

iii. Write three differences between assembly language and HLLs.

Ans: Difference between high-level language and assembly language:

Features of high-level language (HLLs):

i. High-level languages are easily understandable.
ii. The programs that are developed in high-level languages are portable.
iii. In case of high-level languages debugging of the code is easy and the program written is not machine dependent.

Features of Assembly language:

i. Although Assembly level language are not easy to understand they are relatively easier as compared to machine level language.
ii. The programs written in this language are not portable and the debugging process is also not very easy.
iii. The programs developed in assembly language are thoroughly machine dependent.

iv. Write four characteristics of HLLs.

Answer: Characteristics of High-level Languages:

High-level languages have the following characteristics:

1) These languages were developed to make computer programming simple, easier and less prone to errors.
2) High-level languages are not machine dependent. They enable programmers to write programs that are independent of a particular type of computer.
3) Programs written in high-level languages must be translated into machine language by a compiler or an interpreter before execution by the computer.
4) The process of finding and removing errors in programs (debugging) is easier in high-level languages compared to low level language.
5) High-level language programs are highly structured. They allow programmers to break lengthy programs into a number of modules which

can be written and tested independently. This makes writing and testing of programs easier.

v. Define Integrated Development Environment (IDE).

Ans: Integrated Development Environment (IDE):

IDE is computer software that brings all the processes and tools required for program development into one place. IDE's aim is to make the life of programmers easier by grouping together all the tasks needed for building applications into one environment.

Today's modern IDEs have user-friendly Graphical User Interface (GUI). Most of the new programming languages use Integrated Development Environment (IDE) to create, compile and run programs.

vi. Differentiate between constant and variable.

Ans: Constants and Variables:

Constants and variables are used in expressions in computer programs. After an expression is evaluated, the result of the expression is also stored in a variable.

- **Constant:**

  Constants are quantities whose values do not change during program execution.

Types of Constant:

They may be numeric, character or string.

i. Numeric Constants:

Numeric constants are of two types, integer and floating-point numbers.

Integer constants represents values that are counted like the number of students in a class. Some examples of integer constants are 7145, -234, 26 etc.

Floating point constants are used to represented values that are measured like the height of a person which might have a value of 166,75 cm or the weight such as 82.6 kilograms.

ii. Character Constant

Character Constant is one of the symbols in C character set. It includes digits 0 to 9, upper case letters A to Z, lowercase letters a to z, punctuation symbols such as semicolon (;), comma (,), period(.) and special symbols such as +, -, =, >, etc. A character constant is enclosed by single quotes such as 'a', 's' etc.

iii. String Constant:

String Constant contains a string of characters within double quotes such as "Hello Ahmed", "a" etc.

- **Variable:**

A variable is a symbolic name that represents a value that can change during execution of a program. A variable has a name, known as variable name and it holds data of other types. A number or any other type of data held in a variable is called its value. It also indicates the types of value a variable can represent.

Type of Variables:

Variables are of two types, numeric and character.

i. Numeric Variables:

Numeric variables are used to represent numeric values in computer programs. They represent integer and floating-point values. Some examples of numeric variables are sum, length, salary, marks etc.

ii. Character variables:

Character variables represent character values in computer programs. It can represent a single character or a string of characters. Some examples of character variables are name, city, gender etc.

When a variable is used in a computer program, the computer associates it with a particular memory location. The value of a variable at any time is the value stored in the associated memory location at that time. Variables are used so that the same space in memory can hold different values at different times.

vii. Which of the following are valid C variables? Give the reason if not a valid variable area, 5x, Sum, net pay, float, _age, else, case, size 22, my weight

Ans: Valid: -my weight, area, size 22, Sum

Not valid:

5x	A variable can't be start with number
Net pay	Space can't be used in variable name
Float	Data types can't be used as variable name
_age	Special character (_) can't be used in variable name
Else	Commands cannot be used as variable name
Case	Command cannot be used as variable name

Reason: Space not allowed in variable names.

Special characters and numbers are not allowed start the name of variable

Reserve words are not allowed

viii. What are reserved words? Why they should not be used as variable names?

Ans: Reserved Words:

The words that are part of programming language and have special purposes in computer programs are called reserved words or keywords.

Reserved words have predefined use and cannot be used for any other purpose. Reserved words are always in lowercase. There are 32 words defined as reserved words in C. A complete list of reserved words used in C language is given in table.

List of reserved words of C:

Auto	Double	Int	Struct
Break	Else	Long	Switch
Case	Enum	Register	Typedef
Char	Extern	Return	Union
Const	Float	Short	Unsigned
Continue	For	Signed	Void
Default	Goto	Sizeof	Volatile
Do	If	Static	while

ix. Why comments are used in programs?

Ans: Comments in C Language:

It is a good programming practice to add comments in program to make it easy for others to understand it. Comments in the source code are ignored by the compiler.

Comments are added in programs when a fact is necessary to be brought to the attention of program's reader. Generally, programmers write comments at the beginning of the program explaining the reader what the program is intended to achieve and inside the program code where something is to be clarified about the structure of the program.

There are two types of comments

- Single line comment
- Multiple line comment

x. What Is the use of typecasting in C programs?

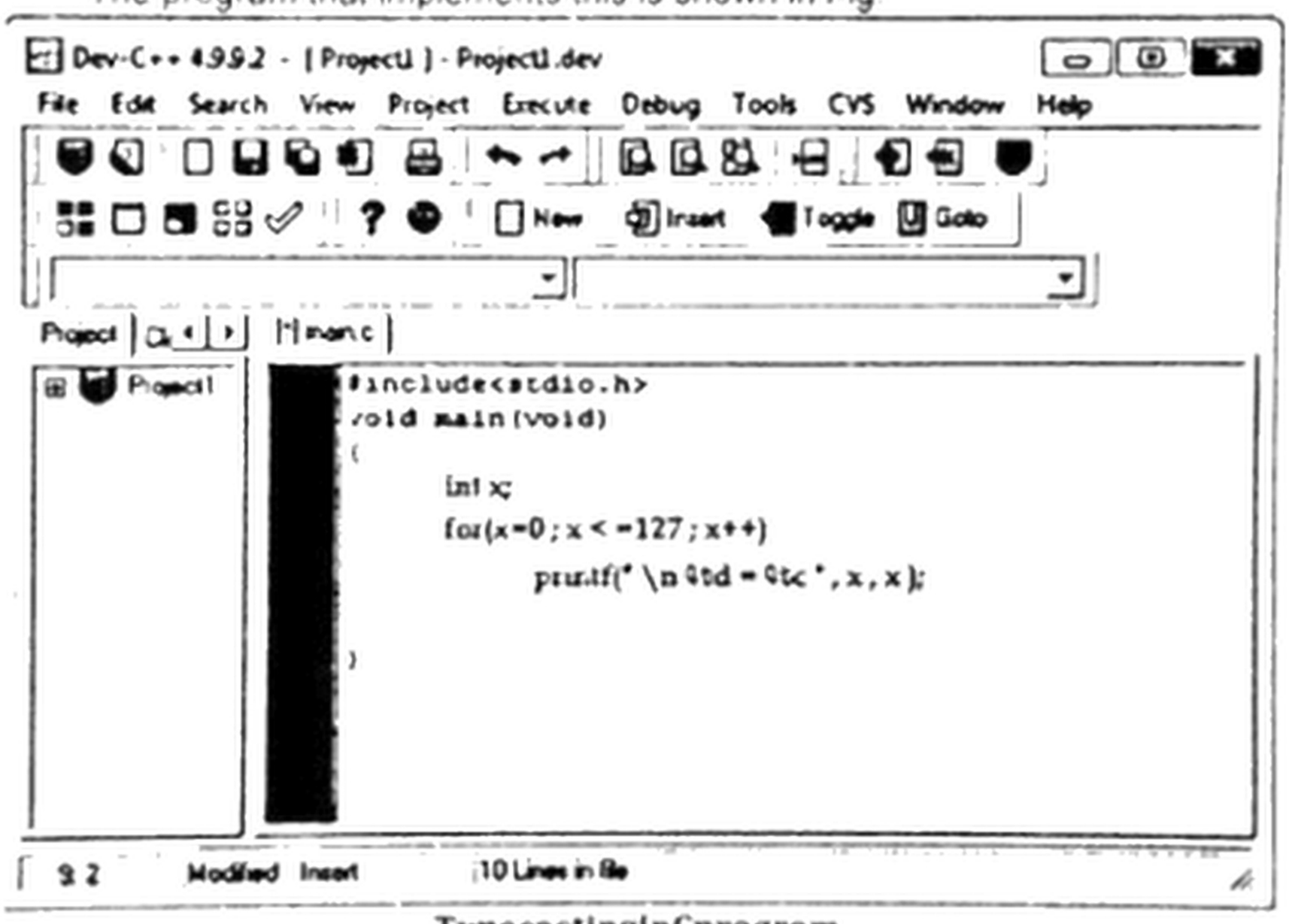
Ans: Type casting in C Language:

Typecasting is a method to convert a variable from one data type to another data type during a program execution. It makes a variable of one type to act like another type.

For example, a variable of type int can act as a variable of type char using typecasting. This required in some situations in programming in C language.

For example, one use for typecasting is when there is need to produce ASCII characters for decimal codes (0 to 127). To do this, the programmer will need to use to typecast to print out the integer variable as its character equivalent using formal specifier %c.

The program that implements this is shown in Fig.



TypecastinginCprogram

There are two types of typecasting in C.

- Implicit typecasting
- Explicit typecasting

EXTENSIVE QUESTIONS

Q3. Describe the following HLLs.

a) C/C++

b) Visual Basic

c) C#

d) Java

Ans: Popular High-level Languages (HLLs):

a) C/C++

C language was developed in early 1970s by Dennis Ritichie at Bell Laboratories. C has become one of the most popular programming languages today. It is a highly structures programming language that is easy to understand and use. In the past, it was mainly used for writing system programs such as operating system, compilers, assemblers, etc.

Today, it is used for writing all type of application programs as well, such as word-processing programs, spreadsheet programs, database management system, educational programs, games etc.

C++ was developed by Bjarne Stroustrup also at Bell Laboratories during 1983-1985. C++ is a superset of C, meaning that any valid C program is also a valid C++ program. The purpose of developing C++ was to provide programming facilities to easily and quickly write more powerful programs.

b) Visual Basic:

Visual Basic (VB) is a high-level language which evolved from the earlier version called BASIC. BASIC stands for Beginner's All-purpose Symbolic Instruction Code. VB is a very popular programming language for writing Windows and Web applications. It provides a graphical development environment to programmers to develop powerful Windows and Web applications.

VB is commonly used for developing business programs such as payroll system and inventory control program. The user can also write programs related with engineering, science, arts, educations, games etc.

c) C#:

C# is a language developed in 2000 by Microsoft Corporation. It is a simple, modern, general purpose programming language.

Syntax of C# is very similar to C and C++. It also has some features of Java. It is a language that makes computer programming easy and efficient. It provides facilities to write Web applications that can be used across the Internet.

All types of programs including games, utilities, operating systems, compilers, business applications and Web based applications can be developed in C#.

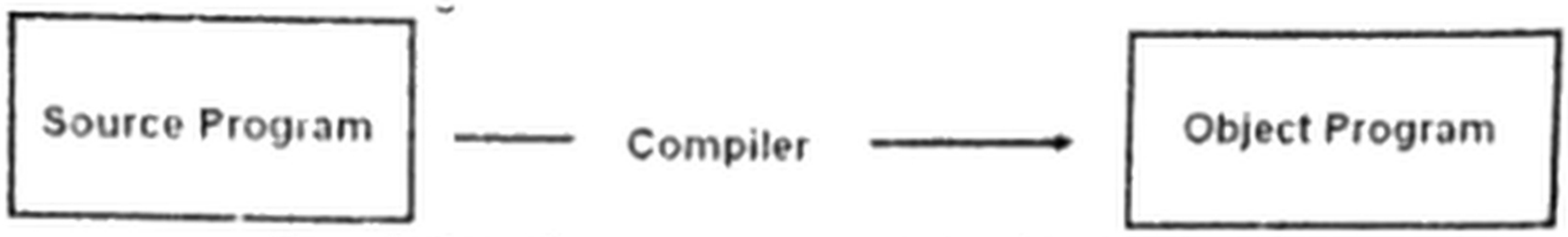
d) Java:

Java is a high-level language developed by Sun Microsystems. It is very similar in syntax to C and C++. In Java, the user can write all types of programs as those written in other programming languages and small programs that can be embedded in a Web page accessed through Internet. Java is an ideal language for network computing. It is used for writing programs for a wide range of devices, computers and networks. It is widely used in Web Application. The current versions of most of the Web browsers are made Java enabled. A few browsers that support Java and Microsoft's Internet Explorer, Firefox and Mozilla.

Q4. Differentiate between compiler and interpreter.

Ans: Compiler and Interpreter:

A compiler is computer software that translates source program into object program as shown in Fig.



Translation of source program into object program

Translation of source program into object program

Source program consists of statements written in a high-level language such as C, Pascal, Java etc.

For example, a program written in C language by a programmer to print table of a number is known as source program. When it is translated with a compiler into machine language, the resulting program is known as object program. The object program is understandable by computer processor but difficult for a human to read and understand because it consists of zeroes and ones.

Interpreter:

Interpreter translates high-level language programs into machine language but it translates one instruction at a time and executes it immediately before translating the next instruction.

Examples of programming languages that use interpreter and Java Script, BASIC, Visual Basic and Perl.

Interpreter reads each statement of source program one at a time and determines what it means as it executes it. It means each time a statement is read, it must be translated into machine language before execution. Compiler translates the entire program into object program before execution by computer. Therefore, a compiled program runs faster.

Q5. Describe the functions of liker and loader programs.

Ans: Linker:

Linker is a software that translates object program into a single executable program. During this process, if it could not find the definition of particular function that is used in the program, then it would assume that it is defined in C library. It will replace this function in the object program with the code from C library and then create a single executable program.

Loader:

It is a software that loads programs into memory and then executes them.

Q6. What are the rules for specifying a variable name in C language?

Ans: Rules for Specifying Variable Names in C language:

The following are the rules for specifying variable names in C language

- A variable begins with a letter or underscore (_) and may consist of letters, underscores and/or digits
- The underscore may be used to improve readability of the variable name For example, overtime
- There is no restriction on the length of a variable name. However only the first 31 characters of a variable are significant. This means that if two variables have the same first 31 characters, they are considered to be the same variables.
- Both upper- and lower-case letters are allowed in naming variables. An upper-case letter is considered different from a lower-case letter. For example, the variable AVG is different from Avg or avg.
- Special characters cannot be used as variable name, e.g #, ?, @ etc
- Reserved words of C languages such as int, case, if etc cannot be used as variable names
- There must be no embedded blank in the name of variable. For example, ma ss is not correct.

Q7. What is the difference between implicit type casting and explicit type casting? Give examples.

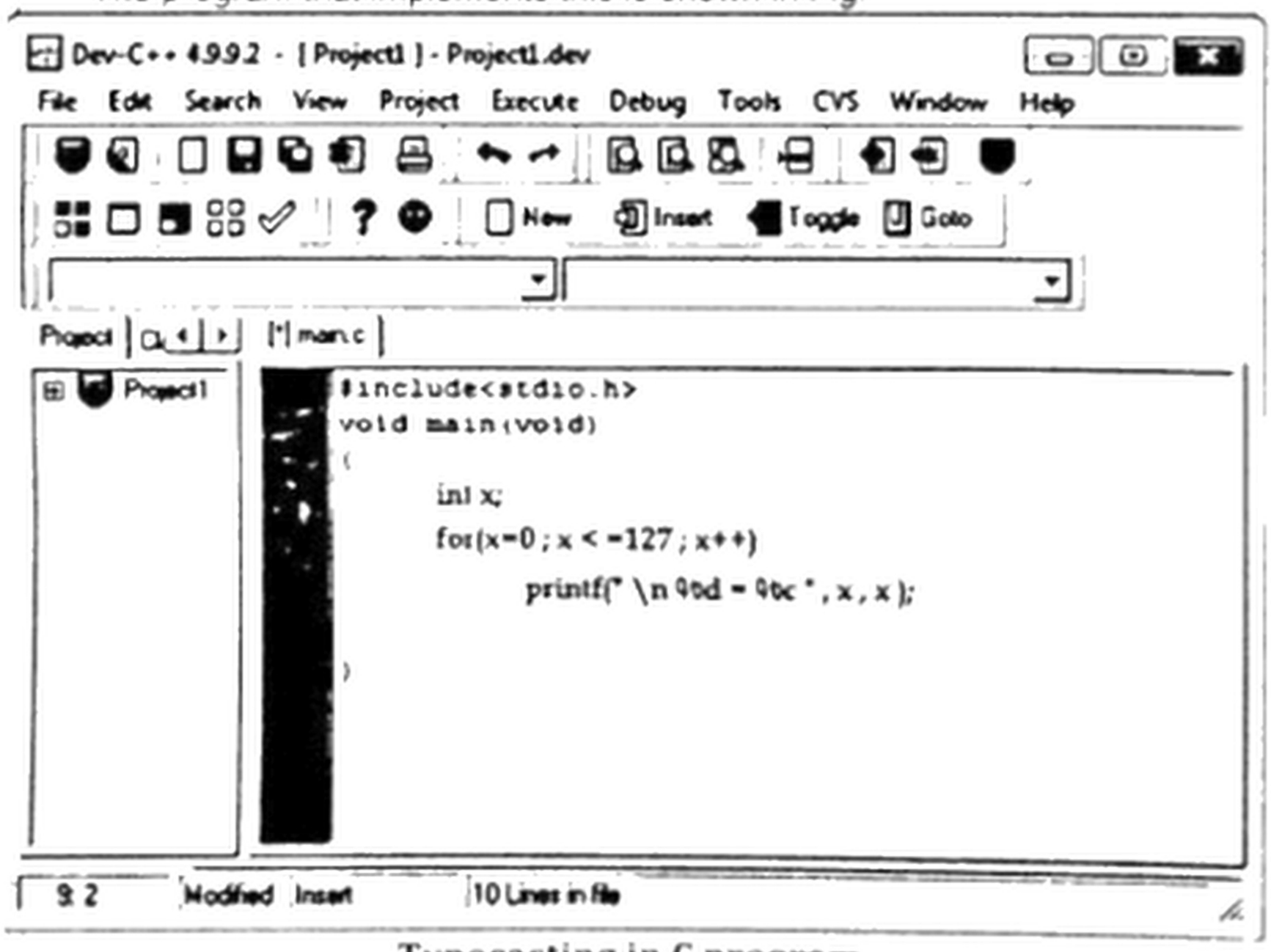
Ans: Typecasting in C Language:

Typecasting is a method to convert a variable from one data type to another data type during program execution. It makes a variable of one type to act like another type. For example, a variable of type int can act as a variable of type char using typecasting. This is required in some situations in programming in C language.

For example, one use for typecasting is when there is need to produce ACSII characters for decimal codes (0 to 127). To do this, the programmer will need to use to typecast to print out the integer variable as its character equivalent using format specifier %c.

The program that implements this is shown in Fig.

The program that implements this is shown in Fig.



Typecasting in C program

There are two types of typecasting in C

- Implicit typecasting
- Explicit typecasting

Implicit Typecasting

Implicit Typecasting is performed automatically by the compiler without programmer's intervention. In this type of casting the compiler converts all operands into the data type of the largest operand. Suppose an expression contains variables of integer and double data types, the result will be evaluated to a double data type.

For example, in the program given below, compiler converts the sum of two float numbers to an int.

```
#include <stdio.h>
  Void main(void)
{
  Float value 1= 2.5;
  Float value2= 5.3;
   Int result
   Result = value1 + value 2;
   Printf("Result: %d", result);
 }
```

The program in Figure is also an example of implicit typecasting.

Explicit Typecasting:

Explicit casting is performed by programmer. The programmer explicitly defines the data type in parenthesis before the variable or expression, as follows:

(type) expression

For example, in the program given below, the programmer converts the division of two int numbers to float.

```
#include <stdio.h>

 Void main(void)

{

Int value1= 30;

Int vlaue2= 7;

Float result;

  Result = (float) value 1/ value 2;

  Printf ("Result  %f" , result);

}
```

Q8. What is a preprocessor? Give examples.

Ans: Preprocessor Directives:

Preprocessor directives are instructions for the C compiler. Every C language program contains certain preprocessor directives at the beginning of the program. Before translating a C language program into machine language, the compiler of C language carries out the processor directives. These directives start with number sign (#)

The most commonly used preprocessor directives are including and define

- **This include Preprocessor Directive:**

It has the following syntax

#include<header file name>

When the preprocessor is carried out by the C compiler, it will search for the header file that is written within the less than (<) and greater than (>) symbols and copy into the source file.

In the above program the header file stdio.h is used. It tells the C compiler to copy the **stdio.h** header file into the program. The **stdio.h** header file stands for standard input-output header. It includes the standard **printf()** and **scanf()** function prototypes. In the above program the printf() functions is used. Therefore, it is required to include this header file in the include preprocessor directive.

- **The define Preprocessor Directive:**

The define preprocessor is used for defining constants (i.e Symbolic constant) in C programs. It directs the compiler to replace all the occurrences of one or more variables in a program with the specified constant. The purpose of using the directive is to give a meaningful name to a constant value that is to be used in the program.

It has the following syntax

#define SYMBOL value/expression

Here, SYMBOL is a valid C variable name and it is by convention written in uppercase. An examples of define preprocessor directive is given below.

#define HEIGHT 12

This preprocessor directive tells the compiler to replace all the occurrences of the variable Height with the value 12, in the program during its execution.

An example of define preprocessor directive using an expression is given below.

#define CUBE (a) (a*a*a)

It can be used in a program as given below

Volume=CUBE(side);

The volume will be calculated as

Volume=(side*side*side);