h - header file.

void - Returns nothing

①

```
# This program will print my name.
# include <stdio.h>
  {
    Printf ("Ampeire Babrah")
     }
```

②

~~cor~~ comment /* */

```
// This program will print my registration number
     and age
   # include < stdio . h >
     {int main ( ) {
  printf (" 2024BSE019");
  printf ("21");
   Return 0;      }
```

printf(" Hellow class\n");

Softwares to use

BORLAND C++

CODEBLOCKS

TERMINAL / COMMAND PROMPT

CS CamScanner

Operators

1 Addition (+)
2 Subtraction (-)
3 Multiplication (*)
4 Division (/)
5 Floor Division (//)
6 Modulus (%)
7 Exponential (**)

```
//A program to use constants
 # include <stdio.h>
 # define PI 3.14159
  const int MAX_LENGTH = 100;
  int main () {
  double radius = 5.0;
  double area = PI * radius * radius ;
  char name [MAX-LENGTH];
 printf ("The area of a circle with radius %.2f
          is %.2f \n", radius, area);
 printf ("The name is : %s\n", name);
  return 0;
  }
```

```
// declation
 float area;       // Initialisation
                      area = 63;
```

float

integer

float length

3

```
# include <stdio.h>
  int main () {
  ~~printf~~ ("Enter your Age ");
  Scanf ("%d ", $ age);
  printf ("My age is %d", age);
      return 0;
     }
```

4

```
// A program to use constants
 # define PI 3·14159  / #define P I 3·1459
   int main () {
   printf ("
```

The C program to tell the size of variable types in bytes

5

```
// This program will multiply two numbers.
# include < stdio.h >
 int main () {
int age = 13;
int height = 4;                        is %d
printf ("Product of %d and %d", age, height; age*
      height);   "Product %d and %d\n is %d", age, height, age*
                                                    height);
return  0;
        }
```

correcti

```
printf ("Product of %d and %d is %d", age, height, age *
                                           height);
```

```
// This is a practice program.
# include < stdio.h >
# define BONUS 2000
 int main ( ) {
// declare a variable (Syntax datatype variable_name;)
    float earning;
// intitialisation
    earning = 10000 ;
// Expected output = You earn 10000
    printf ("You earn %f", earning);
    return 0;
```

CS CamScanner

7

```
# include < stdio.h >
# define  BONUS  2000      ~~#~~float tax;
int main () {
 float earning
 printf ("Please enter ur earning for amonth:");
 scanf ("%f ", &earning);
 earning = BONUS + earning;
 printf ("The earnings including Bonus for a month are
          %.2f", earning);
p ~~BONUS~~ TAX = 100;
 printf ("The earnings after reduction of tax is %f",
          earnings - tax );
 return 0;
     }
```

Name
Reg no

8

```
// program to find the circumference.
# include <studio.h>
# define PI 3.14
int main () {
float rad, circ ;          {Varrables}
printf ("Enter the radius:");
scanf ("%f", $rad);
   circ = 2 *PI *rad;
printf (" The circumference is %f", circ);

  return 0;
      }
```

Decrement (--)   Modulus → 2%2 ⇒ 0 – is the modulus

Increment –(++)   x += 2 or x = x+2

8

The program about operators

```
# include <stdio.h>
int main () {
Int m;
printf ("Enter any integer value");
scanf ("%d", $m);
  m+1;
printf ("%d", M);          {decrement m by 2}
 m-=2                                          --2
printf ("%d", m);          Asign Y to M
 ~~M=Y~~ Y = M;
printf ("%d", Y);
   decrement Y by 1  [ ~~m~~-1 ;  y-1 ;
printf ("% d", Y);
   return 0;
  }
```

++x ⇒ x = x+1, Implement x then use it.   After

x++, ~~Impl~~ Uses x the way it is}   Before

\t - for spaces

10

```
// The program that uses loops and flow control.
# include <stdio.h>
int main () {
~~#~~ int x ;
  int y ;
 if (x = = y) {
  printf ("The valuables are equal:"); }
 printf ("This demonstrates a for loop");
    for ( x = = 10 ; x < 20 ; x --)
      {
     printf ("%d", x );
     printf ("This demonstrates a while loop");
     while ( y ≤ 20 )
      {  y ++ ;
       printf ("%d", y ); }
 // This demonstrates a do while statement.
  do
  printf ("The value is %d", x);
  while (x ≤ 20);
 ~~if (x ≠ y)~~ .
  if (x != y) {
```

CS CamScanner

```
      {
r
ontinue}  break ;  }
for(;)
      else
      if (Y != X) {
printf ("The values are different") }
step 1 :
Printf ("This is the first step");
step 2 :
printf ("This is the second step);
printf ("Enter a value to move to a step");
scanf ("%d", $Z );
if (Z = = 1)
go_to 1 ;
else if ( Z = = 2)
goto step 2 ;
      else
   printf ("Enter either 1 or 2");
return 0;
          }
```