COS1511

Preprocessors

cstdlib

- rand() /srand()
- Atoi / Atol / Atof
- to.string

iomanip

- setw
- set.precision
- showpoint

iostream

cout & cin

string

- getline

cctype

- isalpha /alnum /digit
- is lower/upper /space
- is punct
- to upper /lower

cstring

- strlen
- strcat
- strcpy
- str str

Errors

syntax error

run-tim error → crash & error while program running

logical error

Variable

Data types

int

float

double

long double

string

const → value can't change

char → char +1 = next letter
→ char (i)

bool →

```
bool result;
result = num1 > num2;

if (result)
   "True"
else
   "False"
```

Input

cin

cin.get(); → Values seperated by <enter>

cin.ignore( ); → Inputs a single character → eg. cin.ignore(20, '\n');
↳ deletes enter after 20 characters

getline(cin, name, '\n')
↳ not necisary

Extras

% → modus / remainder

+= →

n++ → add after

++n → add before

Manipulation

string manipulation

Substring → String.substr (start, length)

find → String. find (substring)

insert → String.insert (insertPos, substring)

erase → String. erase (start, length)

replace → String. replace (start, length, substring)
↳ of substring to replace!

precision → cout. precision ( );

fixed point → cout. setf (ios ::fixed);

word size → word. size ( );

- average → avg = float (a+b+c)/3 ;
- Swap → swap (x,y)
- power → pow = (length, 2);

- for (int i=0; i<word. size(); i++)
    cout << word [i];

- for (int i = word.size ( )-1; i>=0 ; i--)
    cout << word [i]

full name = Name + " " + Surname;

Escape sequences

\n → Newline → Print on next line

\t → Tab → Cursor skips over tab stop

\b → Backspace → Cursor backs up

\r → Return → Cursor beginning to line

\a → Alarm → Computer beeps

\\ → Backslash → prints \

\' → SQuote → prints '

\" → DQuote → print "

Loops

While loop

```
while (condition)                    *exit : OPPOSITE
{ statement;
}
```

Do, while loop

```
do
{ statement;
} while (condition);
```

For loop

```
for (condition)             * (int i=1; i<=10 ; i++)
{statement;
}
```

```
for (condition)
{   for (condition)
      {statement;
      }
}
```

Statements

If statements

```
if condition)
 { Statement;
 }
else
 { statement;
 }
```

Switch statement

```
int choice
switch (choice)
            ↳ ordinal (not float/string)
{ case 1:
     statements;
     break;
   case 2: ~
   case 3: ~
   case 4: ~
}
```

```
⊛ char choice
  switch (choice)
   case 'y': ~
```

Class

```
class CName
  { public:
       ~

     private:
       ~

  };
```

Struct

```
Struct   SName
    { int  Variable;
    };

void  Input  (sName  &  sVariableP)
    getline  (cin,  sVariableP. Variable,  '\n')
void  Display  ( const  sName  &  sVariableP)
    {  cout << sVariableP. Variable << end;
    }

int  main  ()
    { SName    sVariable 1;
      Input    (sVariable 1)
      Display   (sVariable 1)
    }
```

Functions

Predefined

```
r = rand ( )                 ⊛ 0-32 767   (always same numbers)
s = srand (  ) ↳ seed
t = srand ( time (0))        ⊛ uses time as seed

10 + rand ( ) % 21           ⊛ Value between 10 & 30
                ↳ values between begin & end
```

writing functions

```
  ↱ function header
        ↱ void if no return type
(return-type   FunctionName   (Parameter List))   ⊛ float x, int y, int z
{ statements;                                ↳ formal parameters
    return   ReturnName;
}


int main ( )
                              ↱ actual parameters
{ cout << FunctionName (Parameter);
    x = FunctionName (Parameter);                   * Func 1 (Func 2 (  ))
}
```

Reference Parameters

```
Void FunctionName (int & Nr)          ⊛ changes to global (number)
 { Statement;
 }


int main ( )
 { FunctionName (number);
 }
```

```
void Func1(int, int);
void Func2(int, int &);
void Func3 (int &, int &);
```

	Main		Function	
	x	y	x	y
`int main()`				
`{  int x,y;`				
`cout<<"Enter number: ";`				
`cin>>x;`	4	?		
`Func1 (x,y);`	4	?		
`Func2 (x,y);`	4	16		
`Func3 (y,x);`	256	16		
`return 0;`				
`}`				
`void Func1 (int x, int y)`	4	?	4	?
`y=x*x;`	4	?	4	16
`void Func2 (int x, int &y)`	4	?	4	?
`y=x* x;`	4	16	4	16
`void Func3 (int &x, int &y)`	4	16	16	4
`y=x*x;`	256	16	16	256

Arrays

One-dimensional

```
int Array [size]
for (int i=0; i<size; i++)
    cin>> Array [i];
```

sorting arrays

```
for (int i=0; i<size; i++)
    { x = rand ( )% maxValue;
      y=0;
      while (y<i && x>=Array [y])
        y++;
      for (int j=i; j>y ; j--)
        Array [j] = Array [j-1];
      Array [y] = x;
    }
```

```
* Void Seat (char Arr [ ][~])
* Seat ( Row Array);
* void Seat (char Arr[ ][~]
```

Arrays as Parameters

```
return-type FunctionName (type ArrayName [], int NumElemsP)
    { statements;
    }
```

2D array

```
const int x = xSize;
const int y = ySize;
int Data [x][y];
```

Transpose

```
void transpose (int matP[ ][N]
  { for (int i=0; i<N-1 ; i++)
        for (int j=i+1; j<N ; j++)
            swap (matP[i][j], matP[j][i]);
  }
```

Array of structs

```
struct   sName
  { Type  Variable;
    Type  Array[size];
  };
void Input (sName & sVariableP)
  { for (int i=0 ; i<size ; i++)
      { cin >> sVariableP. Array[i];
      }
  }
```