

Notes

Introduction to Computer Science (CS50) on EdX

Sparsh Jain

November 16, 2020

Contents

1	Computational Thinking, Scratch	3
1.1	Binary Number System	3
1.2	Algorithms	3
1.3	Time Complexity	3
1.4	Pseudocode	3
1.5	Scratch	3
2	C	4
2.1	Hello World	4
2.2	Input	4
2.3	Initialization	6
2.4	Increment	6
2.5	Conditionals	6
2.6	Loops	6
2.6.1	While Loop	6
2.6.2	For Loop	7
2.7	Additional Info	7
2.7.1	Datatypes	7
2.7.2	Functions	7
2.7.3	Placeholders	8
2.7.4	Arithmetic Operations	8
2.8	Examples	8
2.8.1	Arithmetic	8
2.8.2	Conditional	11
2.8.3	Logical	12
2.8.4	Loop	13
2.8.5	Function	14
2.9	Limitations	19

List of Programs

2.1	Hello World in C	4
2.2	Hello User in C	5
2.3	int.c	9
2.4	float.c	9
2.5	parity.c	10
2.6	conditions.c	11
2.7	agree.c	12
2.8	cough0.c	13
2.9	cough1.c	13
2.10	cough2.c	14
2.11	cough3.c	15
2.12	positive.c	16
2.13	mario0.c	16
2.14	mario2.c	17
2.15	mario8.c	18
2.16	floats.c	19
2.17	overflow.c	19

Chapter 1

Computational Thinking, Scratch

1.1 Binary Number System

1.2 Algorithms

1.3 Time Complexity

1.4 Pseudocode

1.5 Scratch

This was only an introductory lecture. [Click here](#) for more details.

Chapter 2

C

2.1 Hello World

```
1  #include <stdio.h>
2
3  int main(void)
4  {
5      printf("Hello, World!\n");
6  }
```

Program 2.1: Hello World in C

Remark. Need to compile using a compiler like `clang` or `gcc`.

2.2 Input

Remark. In case of errors in compiling, start by trying to *fix* the first one, and so on.

Remark. Use `-lcs50` to link `cs50.h` header.

Remark. Use `make` to ease your life compiling!

```
1  #include <cs50.h>
2  #include <stdio.h>
3
4  int main(void)
5  {
6      string answer = get_string("What's your name?\n");
7      printf("Hello, %s!\n", answer);
8  }
```

Program 2.2: Hello User in C

2.3 Initialization

```
int counter = 0;
```

2.4 Increment

```
counter = counter + 1;  
counter += 1;  
counter++; // Syntactic Sugar
```

2.5 Conditionals

```
if (x < y)  
{  
    printf("x is less than y!\n");  
}  
else if (x > y)  
{  
    printf("x is greater than y!\n");  
}  
else // if (x == y)  
{  
    printf("x is equal to y!\n");  
}
```

2.6 Loops

2.6.1 While Loop

Infinite Loop

```
while(true)  
{  
  
}
```

Repeat

```
int i = 0;  
while(i < 50)
```

```
{  
    printf("Hello World!\n");  
    i = i+1;  
}
```

2.6.2 For Loop

```
for(int i = 0; i < 50; i += 1)  
{  
    printf("Hello World!\n");  
}
```

2.7 Additional Info

2.7.1 Datatypes

Some of these (like `string`) are implemented in `cs50.h` library.

- `bool`
- `char`
- `double`
- `float`
- `int`
- `long`
- `string`
- ...

2.7.2 Functions

They are implemented in `cs50.h` library.

- `get_char`
- `get_float`
- `get_double`
- `get_int`

- `get_long`
- `get_string`
- ...

2.7.3 Placeholders

- `%c` for `char`
- `%f` for `float`
- `%i` for `int`
- `%li` for `long`
- `%s` for `string`

2.7.4 Arithmetic Operations

- `+`
- `-`
- `*`
- `/`
- `%`

2.8 Examples

2.8.1 Arithmetic

```

1  #include <cs50.h>
2  #include <stdio.h>
3
4  int main(void)
5  {
6      int age = get_int("What's your age?\n");
7      // int days = age * 365;
8      // printf("You are at least %i days old.\n", days);
9      printf("You are at least %i days old.\n", age * 365);
10 }

```

Program 2.3: int.c

```

1  #include <cs50.h>
2  #include <stdio.h>
3
4  int main(void)
5  {
6      float price = get_float("What's the price?\n");
7      // printf("Your total is %f.\n", price * 1.18);
8      printf("Your total is %.2f.\n", price * 1.18);
9  }

```

Program 2.4: float.c

```

1  #include <cs50.h>
2  #include <stdio.h>
3
4  int main(void)
5  {
6      int n = get_int("n: ");
7
8      if (n % 2 == 0)
9      {
10         printf("even.\n");
11     }
12     else
13     {
14         printf("odd.\n");
15     }
16 }

```

Program 2.5: parity.c

2.8.2 Conditional

```
1 // Conditions and relational operators
2
3 #include <cs50.h>
4 #include <stdio.h>
5
6 int main(void)
7 {
8     // Prompt user for x
9     int x = get_int("x: ");
10
11     // Prompt user for y
12     int y = get_int("y: ");
13
14     // Compare x and y
15     if (x < y)
16     {
17         printf("x is less than y\n");
18     }
19     else if (x > y)
20     {
21         printf("x is greater than y\n");
22     }
23     else
24     {
25         printf("x is equal to y\n");
26     }
27 }
```

Program 2.6: conditions.c

2.8.3 Logical

```
1 // Logical operators
2 #include <cs50.h>
3 #include <stdio.h>
4 int main(void)
5 {
6     // Prompt user to agree
7     char c = get_char("Do you agree?\n");
8     // Check whether agreed
9     if (c == 'Y' || c == 'y')
10    {
11        printf("Agreed.\n");
12    }
13    else if (c == 'N' || c == 'n')
14    {
15        printf("Not agreed.\n");
16    }
17 }
```

Program 2.7: agree.c

2.8.4 Loop

```
1 // Opportunity for better design
2
3 #include <stdio.h>
4
5 int main(void)
6 {
7     printf("cough\n");
8     printf("cough\n");
9     printf("cough\n");
10 }
```

Program 2.8: cough0.c

```
1 // Better design
2
3 #include <stdio.h>
4
5 int main(void)
6 {
7     for (int i = 0; i < 3; i++)
8     {
9         printf("cough\n");
10    }
11 }
```

Program 2.9: cough1.c

2.8.5 Function

```
1  // Abstraction
2
3  #include <stdio.h>
4
5  void cough(void);
6
7  int main(void)
8  {
9      for (int i = 0; i < 3; i++)
10     {
11         cough();
12     }
13 }
14
15 // Cough once
16 void cough(void)
17 {
18     printf("cough\n");
19 }
```

Program 2.10: cough2.c

```
1  // Abstraction with parameterization
2
3  #include <stdio.h>
4
5  void cough(int n);
6
7  int main(void)
8  {
9      cough(3);
10 }
11
12 // Cough some number of times
13 void cough(int n)
14 {
15     for (int i = 0; i < n; i++)
16     {
17         printf("cough\n");
18     }
19 }
```

Program 2.11: cough3.c


```

1  // Abstraction and scope
2
3  #include <cs50.h>
4  #include <stdio.h>
5
6  int get_positive_int(void);
7
8  int main(void)
9  {
10     int i = get_positive_int();
11     printf("%i\n", i);
12 }
13
14 // Prompt user for positive integer
15 int get_positive_int(void)
16 {
17     int n;
18     do
19     {
20         n = get_int("Positive Integer: ");
21     }
22     while (n < 1);
23     return n;
24 }

```

Program 2.12: positive.c

```

1  // Prints a row of 4 question marks
2
3  #include <stdio.h>
4
5  int main(void)
6  {
7     printf("????\n");
8 }

```

Program 2.13: mario0.c

```

1  // Prints a row of n question marks with a loop
2
3  #include <cs50.h>
4  #include <stdio.h>
5
6  int main(void)
7  {
8      int n;
9      do
10     {
11         n = get_int("Width: ");
12     }
13     while (n < 1);
14     for (int i = 0; i < n; i++)
15     {
16         printf("?");
17     }
18     printf("\n");
19 }

```

Program 2.14: mario2.c

```

1  // Prints an n-by-n grid of bricks with a loop
2
3  #include <cs50.h>
4  #include <stdio.h>
5
6  int main(void)
7  {
8      int n;
9      do
10     {
11         n = get_int("Size: ");
12     }
13     while (n < 1);
14     for (int i = 0; i < n; i++)
15     {
16         for (int j = 0; j < n; j++)
17         {
18             printf("#");
19         }
20         printf("\n");
21     }
22 }

```

Program 2.15: mario8.c

2.9 Limitations

```
1  // Floating-point arithmetic with float
2
3  #include <cs50.h>
4  #include <stdio.h>
5
6  int main(void)
7  {
8      // Prompt user for x
9      float x = get_float("x: ");
10
11     // Prompt user for y
12     float y = get_float("y: ");
13
14     // Perform division
15     printf("x / y = %.50f\n", x / y);
16 }
```

Program 2.16: floats.c

```
1  // Integer overflow
2
3  #include <stdio.h>
4  #include <unistd.h>
5
6  int main(void)
7  {
8      // Iteratively double i
9      for (int i = 1; ; i *= 2)
10     {
11         printf("%i\n", i);
12         sleep(1);
13     }
14 }
```

Program 2.17: overflow.c

[Click here for more examples.](#)