MA144: Problem Solving and Computer Programming

Lecture-11

char, string, errors, break

The Type char

A variable of type char can hold any single character on the keyboard.

Syntax char sym, letr;

```
#include<iostream>
using namespace std;

int main()
{  char sym='A', lettr='a', b=' ';
cout<<sym<<lettr <<endl;
cout<<sym<<b<<lettr;
return 0;
}
```

```cpp
#include<iostream>
using namespace std;

int main()
{  char sym='A', lettr='a', b=' ';
cout<<sym<<lettr <<endl;
cout<<sym<<b<<lettr;
return 0;
}
```

Reading input using cin

```
#include<iostream>
using namespace std;

int main()
{   char sym;
cout<<"enter a character" <<endl;
cin>>sym;
cout<<"the entered character is "<<sym;
return 0;
}
```

```
enter a character
N
the entered character is N
```

```
enter a character
nitw
the entered character is n
```

```
#include<iostream>
using namespace std;

int main()
{   char sym, let;
cout<<"enter a character" <<endl;
cin>>sym>>let;
cout<<"the entered character is "<<sym<<" "<<let;
return 0;
}
```

```
enter a character
nitw
the entered character is n i
```

```
#include<iostream>
using namespace std;

int main()
{  char sym="A", let='a';

cout<<"the characters are "<<sym<<" "<<let;
return 0;
}
```

[Error] invalid conversion from 'const char*' to 'char' [-fpermissive]

Syntax error or **compile-time error**

- string constants are placed inside of double quotes, while constants of type char are placed inside of single quotes
- 'A' and "A" mean different things
- 'A' is a value of type char
- "A" is a string of characters

```
#include<iostream>
using namespace std;

int main()
{  int sym='A', let='a';

cout<<"the characters are "<<sym<<" "<<let;
return 0;
}
```

```
#include<iostream>
using namespace std;

int main()
{  int sym='A', let='a';

cout<<"the characters are "<<sym<<" "<<let;
return 0;
}
```

```
the characters are 65 97
```

The ASCII Character Set

Only the printable characters are shown. Character number 32 is the blank.

32		56	8	80	P	104	h
33	!	57	9	81	Q	105	i
34	"	58	:	82	R	106	j
35	#	59	;	83	S	107	k
36	$	60	<	84	T	108	l
37	%	61	=	85	U	109	m
38	&	62	>	86	V	110	n
39	'	63	?	87	W	111	o
40	(	64	@	88	X	112	p
41	)	65	A	89	Y	113	q
42	*	66	B	90	Z	114	r
43	+	67	C	91	[	115	s
44	,	68	D	92	\	116	t
45	–	69	E	93	]	117	u
46	.	70	F	94	^	118	v
47	/	71	G	95	_	119	w
48	0	72	H	96	'	120	x
49	1	73	I	97	a	121	y
50	2	74	J	98	b	122	z
51	3	75	K	99	c	123	{
52	4	76	L	100	d	124	|
53	5	77	M	101	e	125	}
54	6	78	N	102	f	126	~
55	7	79	O	103	g		

```
#include<iostream>
using namespace std;

int main()
{   char sym='9', let='0';

cout<<"the characters are "<<sym<<" "<<let;
return 0;
}
```

```
#include<iostream>
using namespace std;

int main()
{  char sym='9', let='0';

cout<<"the characters are "<<sym<<" "<<let;
return 0;
}
```

```
the characters are 9 0
```

```
#include<iostream>
using namespace std;

int main()
{  int i=9;
    int x='9';

   cout<<i<<' '<<x;

    return 0;
}
```

Describe error/find output

```
#include<iostream>
using namespace std;

int main()
{   char sym='abcdef', let;

cout<<"the characters are "<<sym<<endl;
cout<<"enter the above string abcdef \n";
cin>>let;
cout<<"entered characters are "<<let;
return 0;
}
```

Errors

- Syntax Errors/compile-time errors
- Runtime Errors
- Logical Errors
- Linked Errors
- Semantic Errors

Syntax Errors/compile-time errors

- occur when the syntax of C++ is violated

```
#include<iostream>
using namespace std;

int main()
{
  cout<<"it is an error"
  return 0;
}
```

[Error] expected ';' before 'return'

Runtime Errors

- occur while the program is executing (compilation will be completed successfully)

```
#include<iostream>
using namespace std;

int main()
{
int a=4,b=0;
cout<<a/b;
return 0;
}
```

Program crashes during runtime.

Logical Errors

- Occur due to poor understanding of the problem and solution procedure
- Even if the syntax is correct, we may **not** get the desired output due to logical issues

```
#include<iostream>
using namespace std;

int main()
{
int i;
  for(i=1;i<=10;i++);
  {
    cout<<"wel come";
  }
return 0;
}
```

We have written this program to print the string “wel come” 10 times, but it prints only once, due to logical issue.

Linked Errors

- Occur when the program is successfully compiled and attempting to link the different object files with the main object file

Semantic Errors

- Occur when it is syntactically correct but has no meaning.

  This is like grammatical mistakes.

```
#include<iostream>
using namespace std;

int main()
{  int a=1,b=2,c;
   a+b=c;
   cout<<c;
   return 0;
}
```

[Error] lvalue required as left operand of assignment

The class `string`

- Although C++ lacks a native data type to directly manipulate strings, there is a `string` class that may be used to process strings in a manner similar to the data types
- To use the `string` class we must include the string library

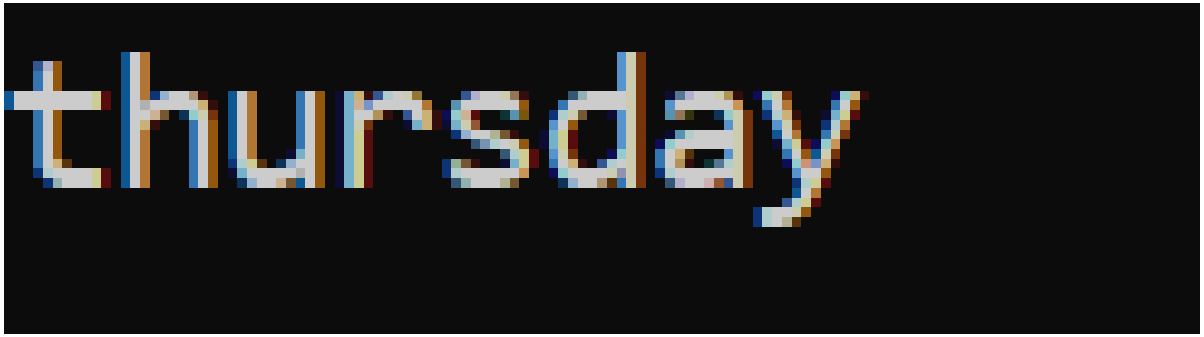
```
#include <string>
```

- Declare variables of type `string` just as you declare variables of other types

```
string day;
day = "thursday";
```

```
#include<iostream>
#include<string>
using namespace std;

int main()
{   string day="thursday";
    cout<<day;
    return 0;
}
```



We use `cin` object to read the strings

```
#include<iostream>
#include<string>
using namespace std;

int main()
{  string name;
    cout<<"enter a string:  ";
    cin>>name;
    cout<<"the entered string is  "<<name;
    return 0;
}
```

```cpp
#include<iostream>
#include<string>
using namespace std;

int main()
{  string name;
    cout<<"enter a string:  ";
    cin>>name;
    cout<<"the entered string is  "<<name;
    return 0;
}
```

```
enter a string:  nitw
the entered string is  nitw
```

```
#include<iostream>
#include<string>
using namespace std;

int main()
{  string name;
    cout<<"enter a string:  ";
    cin>>name;
    cout<<"the entered string is  "<<name;
    return 0;
}
```

```
enter a string:  nit warangal
the entered string is  nit
```

- When we use cin to read input into a string variable, the computer only reads until it encounters a whitespace character
(we cannot input a string that contains spaces)

```
#include<iostream>
#include<string>
using namespace std;

int main()
{  string name, name1;
    cout<<"enter a string:  ";
    cin>>name>>name1;
    cout<<"the entered string is  "<<name<<" "<<name1;
    return 0;
}
```

```
enter a string:  nit warnagal
the entered string is  nit warnagal
```

```
enter a string:  nit warangal ece
the entered string is  nit warangal
```

- Using ‘+’ symbol between two strings,
  we concatenate the two strings together to create one longer string

```
#include<iostream>
#include<string>
using namespace std;

int main()
{  string name, name1="nit", name2="warangal";

    name=name1+name2;

    cout<<"the concated string is  "<<name;
    return 0;
}
```

```
#include<iostream>
#include<string>
using namespace std;

int main()
{  string name, name1="nit", name2="warangal";

    name=name1+name2;

    cout<<"the concated string is  "<<name;
    return 0;
}
```

```
the concated string is  nitwarangal
```

Note that a space is not automatically added between the strings.

```
#include<iostream>
#include<string>
using namespace std;

int main()
{  string name, name1="nit", name2="warangal";

    name=name1+" "+name2;

    cout<<"the concated string is  "<<name;
    return 0;
}
```

```
the concated string is  nit warangal
--------------------------------
```

The `break` Statement (in loops)

- break statement consists of the keyword `break` followed by a semicolon

```
break;
```

- Used to exit a loop
- It ends only the **innermost loop** that contains it.

```
#include<iostream>

using namespace std;

int main()
{  int i, a;
  for(i=1;i<=5;i++)
  {
     if(i==1)
       { cout<<"i am in if block"<<endl;
        break;
         }

       cout<<"hai"<<endl;
  }
  cout<<"welcome";


    return 0;
}
```

```
i am in if block
welcome
```