

## **Designs 2 (Selection)**

**Graded course work**

---

### **Important Notes:**

- **Write all the programs, from the designs in this handout**
- **Use the same program names, and variable names that I have specified**
- **Compile, run, and test your programs**
- **Submit a copy of these programs to me for grading**
- **If you don't know how to do any particular part, ask me -- I will show you how**

**Program (date1.c)**

Write a program that ask the user to enter a number for the month, day, and year, then prints out the date in English.

Extend the program to work with 3 months, 3 days, and 3 years.

---

declare integers "the\_month", "the\_day", "the\_year"

ask for the month number

put the number in the box "the\_month"

ask for the day number

put the number in the box "the\_day"

ask for the year number

put the number in the box "the\_year"

if ("the\_month" is 1)

    tell "January,"

if ("the\_day" is 1)

    tell "the first,"

if ("the\_year is 2000)

    tell "two thousand."

---

**Program (`highest1.c`) Cliché**

In programming, there are many code-fragments that you will use over-and-over again.

I call them clichés, because they were invented long ago by programmers, and other programmers continue to reuse them.

The first cliché is used when you want to keep hold of the highest value so far. This example isn't the full cliché, but it shows you the most important part of it.

---

declare floats "high\_temperature", "highest\_high\_temperature"

put lower than possible value (temperature) into the box "highest\_high\_temperature"

ask for today's highest temperature

put the number in the box "high\_temperature"

if ("high\_temperature" is greater than "highest\_high\_temperature")

    put "high\_temperature" into the box "highest\_high\_temperature"

tell the number "highest\_high\_temperature"

---

**Program (total1.c) Cliché**

This next cliché show you how to keep a running total in a program. Again this is not the full cliché, but it shows you the most important part of it.

Often a grocery store receipt might have a line: "Number of items: 7", this cliché forms the basis of how that's done.

---

declare integer "number\_of\_items"

put some value into box "number\_of\_items" (for this demonstration only)

6

calculate: "number\_of\_items" + 1 (increment the variable) and put the result  
into the box "number\_of\_items"

tell the number "number\_of\_items"

---

---

---

**Program (odd\_even1.c) Cliché**

Write a program that asks for an integer, then tells the user if it is an odd or even number (if an even number is divided by 2, its remainder will be 0). Remember that the remainder operator is the '%'.

---

declare integer "the\_number"

ask for a number

put the number in the box "the\_number"

if (the remainder of "the\_number" divided by 2 equals 0)

tell the number "the\_number" is even

else

tell the number "the\_number" is odd

---

**Program (is\_upper1.c) Cliché**

Write a program that asks for a character to be entered, then tells if that character is an uppercase letter.

The **and** operator needed in this program is (&&).

---

declare character "the\_character"

ask for a character

put the character in the box "the\_character"

if ("the\_character" is between 'A' and 'Z')

    tell that "the\_character" is an uppercase letter

---

---

**Program (validate1.c) Cliché**

Usually in a program, you want to check that the data entered by the user is "good data". You do this by validating (testing) the data.

For example, a number entered for a month should be between 1 and 12, any other value would be "bad data".

---

declare integer "the\_month"

ask for a number for the month

put the number in the box "the\_month"

if ("the\_month" is between 1 and 12)

    tell "Ok -- good value"

else

    tell "Not ok -- bad value"

---

**Program (to\_upper1.c) *Cliché***

Write a program that asks for a lowercase letter, then converts it to its uppercase version. This program is simpler than you might think.

---

declare character "input\_character", "output\_character"

ask for a character

put the character in the box "input\_character"

subtract 32 from "input\_character" and put the result in "output\_character"

tell the value of "output\_character"

---

---

**Program (split1.c) *Extra Credit***

Write a program that will take a 4 digit number, and split it into *thousands*, *hundreds*, *tens*, and *units*.

If the program input is: 1234

The program output should be:

The thousands are: 1

The hundreds are: 2

The tens are: 3

The units are: 4

## Extra Work (Not Graded)

These programs will **not** be graded, but they are worth doing if you have the time.

---

### Program (square1.c)

Write a program that asks for a number, then returns the square (the number multiplied by itself) of that number.

---

declare floats "the\_number", "the\_square"

ask for a number

put the number in the box "the\_number"

calculate: "the\_number" \* "the\_number" and put the result in the box "the\_square"

tell the number in the box "the\_square"

---

---

### Program (lowest1.c) *Cliché*

This program is very similar to the program `highest1.c`, but it keeps hold of the lowest value so far.

---

declare floats "low\_temperature", "lowest\_low\_temperature"

put higher than possible value (temperature) into the box "lowest\_low\_temperature"

ask for today's lowest temperature

put the number in the box "low\_temperature"

if ("low\_temperature" is less than "lowest\_lowest\_temperature")

    put "low\_temperature" into the box "lowest\_low\_temperature"

tell the number "lowest\_low\_temperature"

---

**Program (is\_lower1.c) Cliché**

Write a program that asks for a character to be entered, then tells if that character is a lowercase letter.

---

declare character "the\_character"

ask for a character

put the character in the box "the\_character"

if ("the\_character" is between 'a' and 'z')

    tell that "the\_character" is an lowercase letter

---

---

**Program (is\_digit.c) Cliché**

Write a program that asks for a character to be entered, then tells if that character is a digit.

---

declare character "the\_character"

ask for a character

put the character in the box "the\_character"

if ("the\_character" is between '0' and '9')

    tell that "the\_character" is a digit

---



**Program (to\_lower1.c) Cliché**

Write a program that asks for an uppercase letter, then converts it to its lowercase version.

This program is simpler than you think.

---

declare character "input\_character", "output\_character"

ask for a character

put the character in the box "the\_character"

add 32 to "input\_character" and put the result in "output\_character"

tell the value of "output\_character"

---