

# Debugging & Testing

Topic 11

---

---

---

---

---

---

---

---

## Testing & Debugging Strategies



Topic 9 - Testing & Debugging

2

---

---

---

---

---

---

---

---

## Testing & Debugging

- Testing is as important as coding
  - Buggy code is costly
- Testing and debugging go together like peas in a pod:
  - Testing **finds** errors;
  - debugging localizes and **repairs** them.
  - Together these form the “testing/debugging cycle”: we test, then debug, then repeat.
- Debug then retest!
  - This avoids (reduces) the introduction of new bugs when debugging.

Topic 9 - Testing & Debugging

3

---

---

---

---

---

---

---

---

## Fixing Compiler Errors

- When you get a compiler error...
  - Try to read the message and think about what it is trying to tell you
  - If it says something like "missing ;"
    - Look at the line above
- If you are getting a "binaries" error or can't open output file
  - Got to Project → clean
- As a habit you should close your previous projects
  - Right click on the project folder and go to close project

Topic 11 - Testing

4

---

---

---

---

---

---

---

---

## Make Debugging Easier

- Using proper style helps
  - Making your code more readable makes it easier to find your errors.
  - Little things like spaces between operators and operands
    - Such as... << and >> operators for cout and cin
- Make sure you name your variables with names that make sense
- Insert cout statements and output your variables to make sure they have the values you expect
  - You can delete them after you've found your problem
- Try to isolate the problem
  - Use ctrl-/ to comment out code and ctrl-/ to uncomment it

Topic 11 - Testing

5

---

---

---

---

---

---

---

---

## Code Incrementally

- Don't try to sit down and write the whole program at once → then debug.
- Write one section at a time and test it.
  - For example:

```
cout << "Enter your annual income: ";  
cin >> income;  
  
cout << "Enter your pay increase rate: ";  
cin >> increaseRate;  
  
cout << "TESTING: income: " << income;  
cout << "\tincrease rate: " << increaseRate << endl << endl;
```

Topic 11 - Testing

6

---

---

---

---

---

---

---

---

## Check Dependent Variables

- o If one calculation is dependent upon another check the dependent variables.

```
salesTax = salesTaxRate * retailPrice;  
totalPrice = retailPrice + salesTax;
```

```
cout << "TESTING: sales tax: " << salesTax;  
cout << "\tsales tax rate: " << salesTaxRate;  
cout << "\t retail price: " << retailPrice;  
cout << "\ttotalPrice: " << endl << endl;
```

- o Testing Loops

- Check the LCV and each variable at each iteration
- USE `cin.get()` ← to hold the cursor
  - This way you trace the code through each iteration

Topic 11 - Testing

7

---

---

---

---

---

---

---

---

## Code Walkthroughs

- o Perform a desk check

- Write down all the variables used in the code segment
  - At this point it is best to walk through every line of code
- Don't skip steps
- Don't assume values
  - if there is no value in your variable and you are using it make sure you are initializing it in the code

Topic 11 - Testing

8

---

---

---

---

---

---

---

---

## Testing

- o Before you run your code
  - Write a test plan
- o Think about the **expected inputs** and **expected outputs** → make a chart

Inputs			Expected Outputs	
Name	prevAmt	amtSpent	Name	Pocket Money
Jean Cyr	12.50	23.00	Jean Cyr	9.50
J.R.	23.57	15.00	J.R.	11.43

Topic 11 - Testing

9

---

---

---

---

---

---

---

---

## Also think about boundary tests

- Try 0s or if you have a selection statement
  - (if statement)
- Test the boundaries
  - if (hours > 40)
  - Test these values: 40, 39, 41
- Test for unexpected inputs

Inputs			Expected Outputs	
Name	prevAmt	amtSpent	Name	Pocket Money
Jean Cyr	-10.00	23.00	Jean Cyr	-13.00

Topic 11 - Testing

10

---

---

---

---

---

---

---

---

---

---

## Test your conditional statements

- Test the paths of your code
- If statements
  - test true
  - test false
- Loops
  - While loops
  - what happens if first input causes the loop to exit
- Test combinations

Topic 11 - Testing

11

---

---

---

---

---

---

---

---

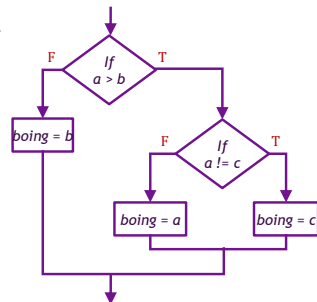
---

---

## Testing Selection Statements

- If a, b, & c are integers.
- What do we test?

a	b	c
1	2	3
1	3	2
2	1	3
2	3	1
3	1	2
3	2	1
1	1	1
1	1	2
1	2	1
2	1	1
1	2	2
2	1	2
2	2	1



What else should we test?

Topic 11 - Testing

12

---

---

---

---

---

---

---

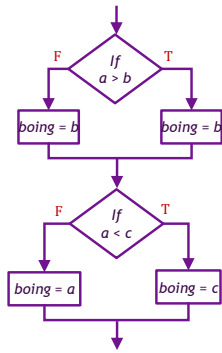
---

---

---

- o If a,b, & c are integers.
- o What do we test?

a	b	c
1	2	3
1	3	2
2	1	3
2	3	1
3	1	2
3	2	1
1	1	1
1	1	2
1	2	1
2	1	1
1	2	2
2	1	2
2	2	1



What else should we test?

---



---



---



---



---



---



---