**CISP 360 Note Sheet Ch 1 – 8**

|  |  |  |
| --- | --- | --- |
| **Major Data types:**  int x = 1;  double x = 1.0;  string s = "";  char c = 'a';  bool b = true;  **Program Header:**  #include <iostream>  using namespace std;  int main()  {  …  }  **Input:**  getline(cin, s);  cin >> x;  **Output:**  cout << x;  **Math functions:**  #include <cmath>  sqrt(x) //square root  pow(base, exponent) //power  fabs(x) //absolute value  **Random Numbers:**  #include <cstdlib>  …  srand(seed);  //random in range [a…b]  x = rand() % (b - a + 1) + a;  **Relational Operators:**  <, >, <=, >=, ==, !=  **Logical Operators:**  &&, ||, !  **Check if x is outside [low...high]:**  !(x >= low && x <= high)  (x < low || x > high)  **Check if x is inside [low... high]:**  (x >= low && x <= high)  !(x < low || x > high)  **Float Comparison**  if(fabs(x – target) < 0.0001)  //… | **if / else if / else:**  if (expr1)  {  …  }  else if (expr2)  {  …  }  ...  else if (exprN)  {  …  }  else  {  …  }  **Switch:**  switch (variable)  {  case value1:  // Statements  break;  case value2:  // Statements  break;  ...  default:  // Statements  break;  }  **While Loop:**  while (expression)  {  // Statements  }  **Do-While Loop:**  do  {  // Statements  } while (loopExpression);  **Postfix increment:** i++  **Prefix increment:** i-- | **For Loop:**  for(initExpression; condition; update)  {  // Statements  }  **Iterate from [0…max - 1]:**  for(int i = 0; i < max; i++)  {  // Statements  }  **break;**  Exits the current loop immediately  **continue;**  Jumps to the next iteration  **String Access**  s[i] s.at(i)  **Streams**  #include <fstream>  **Output File**  ofstream ofs(fileName);  if(!ofs)  cout << "Error opening..";  ofs << x << endl;  **Input File**  ifstream ifs(fileName);  if(!ifs)  cout << "Error opening..";  string s;  while(ifs >> s)  //…  while(getline(ifs,s))  //get an entire line each time  //… |

|  |  |
| --- | --- |
| **Arrays**  type name[size]; //uninitialized  //initialization list  type name[] = {val1, val2, …};  //Element access  //index starts at 0 and ends at size - 1  name[index]  type name[rows][cols]; //2D  //initialization list  type name[][cols] = {{val1,…}, {val2,…},…};  //element access  name[i][j]  //loop to print out a 2D array  //newline after each row, space between cols  for(int i = 0; i < rows; i++)  {  for(int j = 0; j < cols; j++)  {  cout << name[i][j] << ' ';  }  cout << endl;  } | **Functions**  //header  //returnType void if no return value  returnType funcName(type name1, type name2, …)  {  //body  return expression; //variable or literal  }  **Reference Variable**  //header  returnType funcName(**type&** name1, …)  {  //body  return expression; //variable or literal  }  **Function with 1D Array**  returnType funcName(type name[], int size)  {  //…  }  **Function with 2D Array**  returnType funcName(type name[][cols], int rows, int cols)  {  //…  } |