Name: Reno Redaja

Date: September 1, 2020

1. Write a short program that asks the user for an integer. Calculate the square and cube of the integer (x2, x3) and print those values out for the user.

#include <stdio.h>

#include <stdlib.h>

int main(int argc, const char \* argv[]) {

int num; //Variable num to store user integer

printf("Enter a number to be squared & cubed: ");

scanf("%d", &num); //Stores user input

int squared = num \* num; //Variable squared is assigned to user integer input = (num^2)

printf("Your number entered squared is: %d \n", squared); //Prints user input, squared

int cubed = num \* num \* num; //Variable cubed is assigned to user integer input = (num^3)

printf("Your number entered cubed is: %d \n", cubed); //Prints user input, cubed

return 0;

}

1. Explain what SDLC stands for and also explain each section of it.

SDLC stands for Software Development Life Cycle. This is a process that is used by the software industry to design, develop & test high quality software. A typical SDLC consists of 6 stages: 1. Planning & Requirements Analysis, 2. Defining Requirements, 3. Designing the Product Architecture, 4. Building or Developing the Product, 5. Testing the Product, & 6. Deployment in the Market & Maintenance.

Planning & Requirements Analysis: This is where developers of the team receive inputs from the customer, sales department, market surveys, etc. This information will be used to plan the project.

Defining Requirements: Once the requirements analysis is complete, the next step is to make sure the product requirements are approved from the customer through an Software Requirement Specification (SRS) which consists of the product requirements to be designed & developed.

Designing the Product Architecture: The SRS is used by architects to come up with the best product to be developed. There is usually more than one design for the product & is proposed and documented in a Design Document Specification (DDS).

Building or Developing the Product: This is the stage in SDLC where the product is being built. The code is generated according to the DDS. From there the developers must follow the coding guidelines by the organization such as which compilers, interpreters, debuggers, etc. are to be used. As well as what type of programming languages will be used for the product.

Testing the Product: From here the product will be tested for any defects. Any defects will be reported, tracked, fixed and retested until the product reaches its best performance defined by the SRS.

Deployment in the Market & Maintenance: Once the product is tested & ready, it will be released to its appropriate market. After it is released, the product will receive feedback and would have to undergo maintenance to suit the customer’s requirements/expectations.

1. Which of the following are valid identifiers:

a) \_hello valid b) hello valid

c) 1me invalid (cannot begin w/ a number) d) me1 valid

e) const invalid (const is a keyword) f) main valid

g) Dave valid h) DAVE valid

1. Create a short program that asks the user for 2 integers. Create a const called PI and set that to 3.14. Return the sum of the users 2 integers multiplied by the constant PI.

#include <stdio.h>

#include <stdlib.h>

int main(int argc, const char \* argv[]) {

int num1, num2; //Variables num1 & num2

const double PI = 3.14; //Constant float variable given the name PI assigned to 3.14

printf("Enter your first number: \n");

scanf("%d", &num1); //Stores user input first number

printf("Enter your second number: \n");

scanf("%d", &num2); //Stores user input second number

printf("\n"); //prints a new line

int sum = num1 + num2; //Variable sum adds the user input

printf("The sum of the numbers you entered is: %d \n", sum); //Prints sum of input

printf("The sum multiplied by constant PI is: %f \n", sum \* PI); //Prints sum multiplied by constant PI

return 0;

}

1. If x = 10; show what values are after the command (each line is separate):
2. int y = x++; x = 11 y = 10
3. int y = ++x; x = 11 y = 11
4. int y = x % 2; y = 0
5. x+=5; x = 15
6. int y = --(x++) x = y =

COMPILATION ERROR

1. int y = (x--)++ x = y =

COMPILATION ERROR

1. x %= 5 x = 0
2. x \*= 5 x = 50
3. If x = 10, and y = 2.0 show the value of z after the command (each line is separate) :
4. double z = x/y z = 5.000000
5. int z = x / y z = 5